11th Overset Grid Symposium
October 15th-18th, 2012
Dayton, Ohio, USA

Common Aerospace Applications at DLR Utilizing the Overset Grid Capabilites of DLR’s CFD Codes

L. Reimer, Th. Schwarz, A. Raichle, R. Heinrich, F. Spiering, A. Stürmer, S. Crippa, Ph. Kelleners, N. Bier, G. Einarsson, N. Kroll

German Aerospace Center
Institute of Aerodynamics and Flow Technology
Departments „Transport Aircraft“ & C²A²S²E
Braunschweig, Germany

DLR

Outline

- Methodology
  - DLR's solvers w/ overset grid capabilities
  - Our hole-cutting approaches
  - Chimera wall projection
  - Load integration
- Applications
  - CFD-6DoF
  - CROR
  - Helicopters
  - Control surface motion
  - Local discretisation improvement
  - Code-to-code coupling

DLR

Methodology

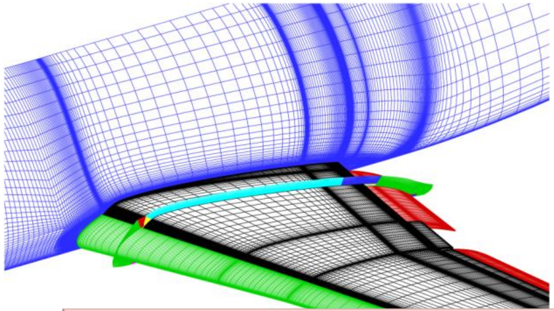
DLR's CFD Solvers w/ Overset Grid Capabilities

Common to both solvers:

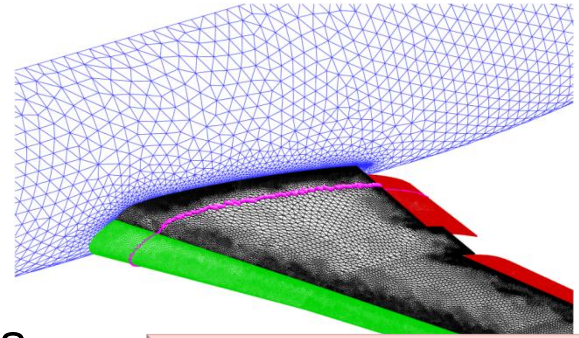
- Developments started in mid 1990s
- FV, 2nd order accuray in space & time
- ALE capability
- Moving, deforming meshes
- Turbulence models: 1-eqn. SA variants, 2-eqn. k-ω-type models, EARSM, RSM; DES, LES
- Fluxes: Central+scalar/matrix diss., Upwinding
- Steady time int.: RK (FLOWer/TAU), LU-SGS, line-implicit (TAU) + multigrid
- Unsteady time int.: Dual-time stepping
- Hybrid MPI/OpenMP-based parallelisation

Special to TAU:

- Work horse of Airbus for aerodynamics analysis on unstructured meshes
- Python interfaces => scripting, in-memory data exchange (6DoF, CFD-CSM coupl.)
- Feature & adjoint-based adaptation



Multiblock-structured compressible RANS solver ***FLOWer***

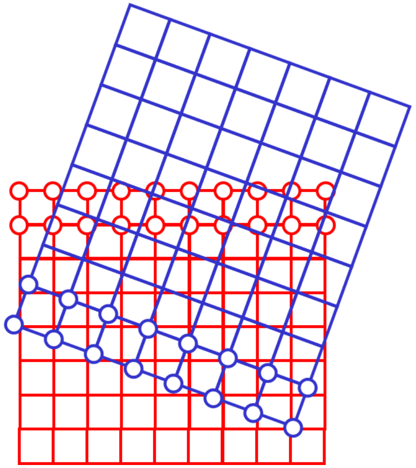


Unstructured compressible RANS solver ***TAU***

DLR

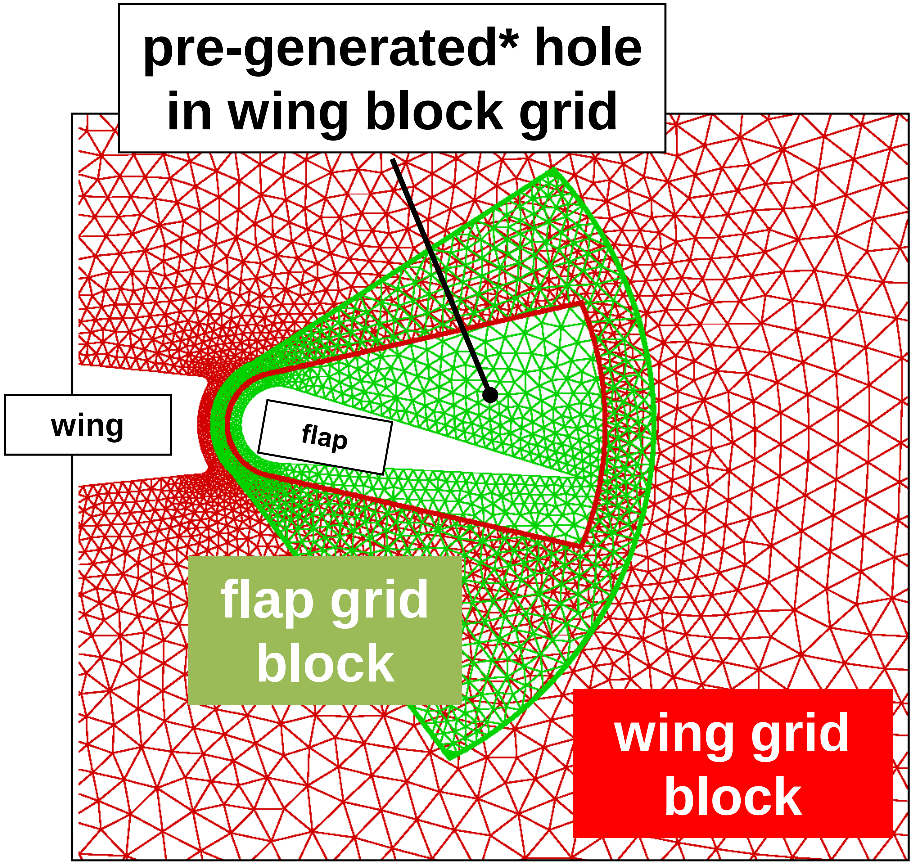
Our Overset Grid Interpolation Technique

- Parallel donor search integrated directly in solver loop (no external tool, no file I/O); 2-stage search
  1. ADT-based search
  2. Iso-parametric mapping based search
- Interpolation is based on primary mesh cells using FE shape functions
  - Only 4 different FE shape functions need to be considered
  - Dual mesh contains arbitrary polyhedrons which would make interpolation more complex & expensive
- Interpolation takes place only on finest multigrid level
- 2 fringe layers

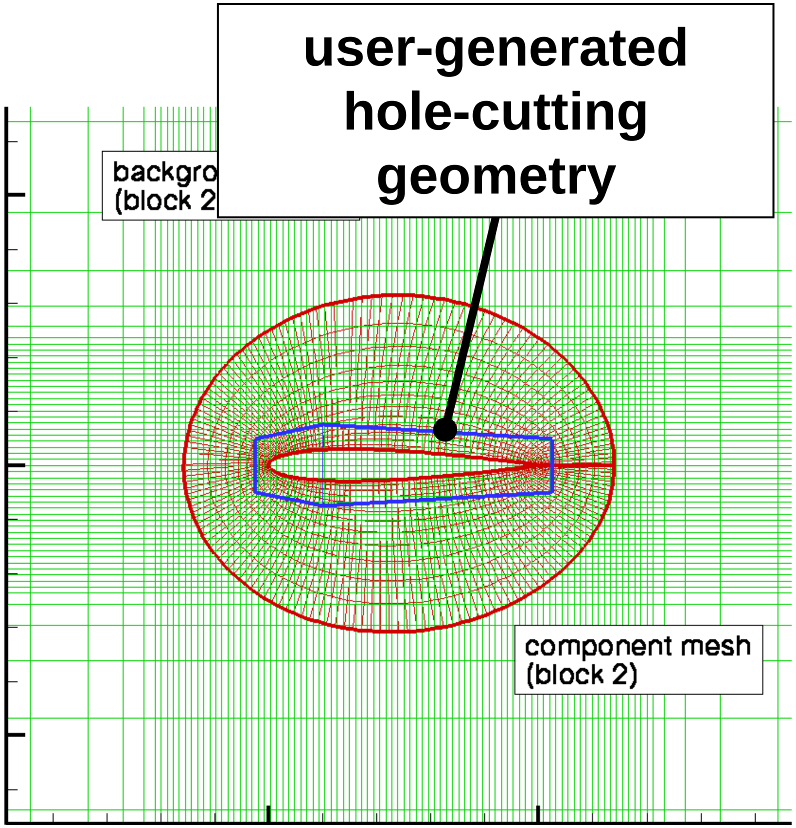


Our Hole-Cutting Strategies

NO hole-cutting



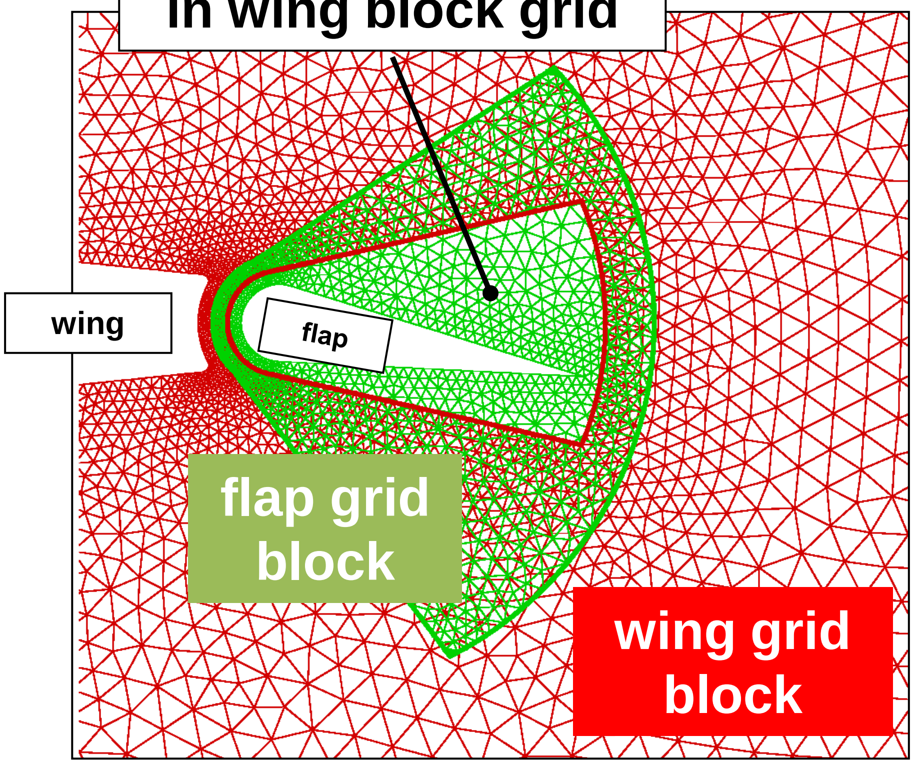
Semi-automatic hole-cutting



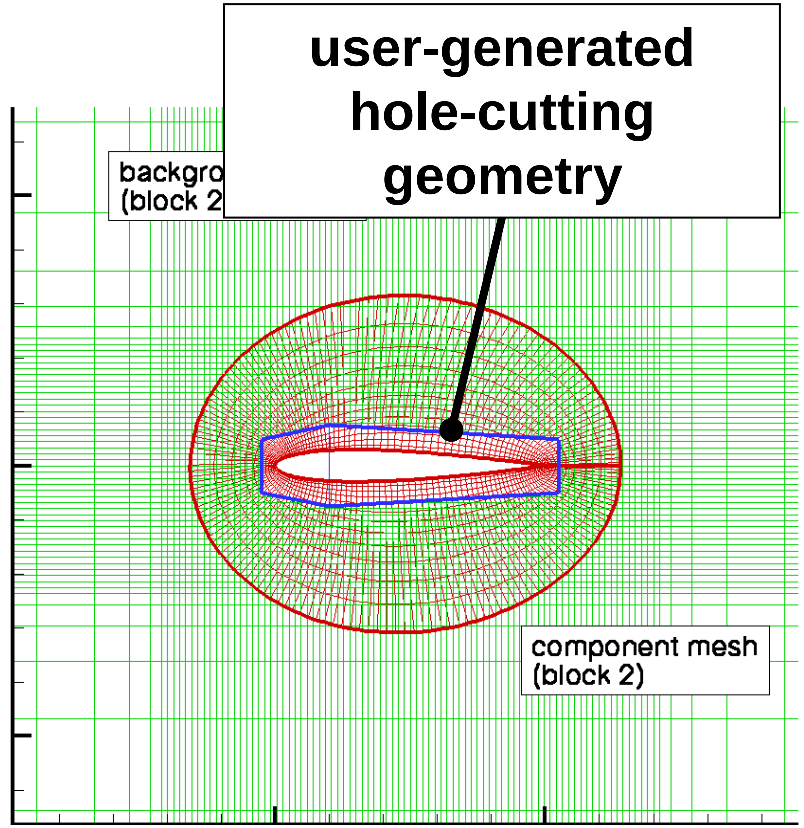
*generated during mesh generation

Our Hole-Cutting Strategies

NO hole-cutting



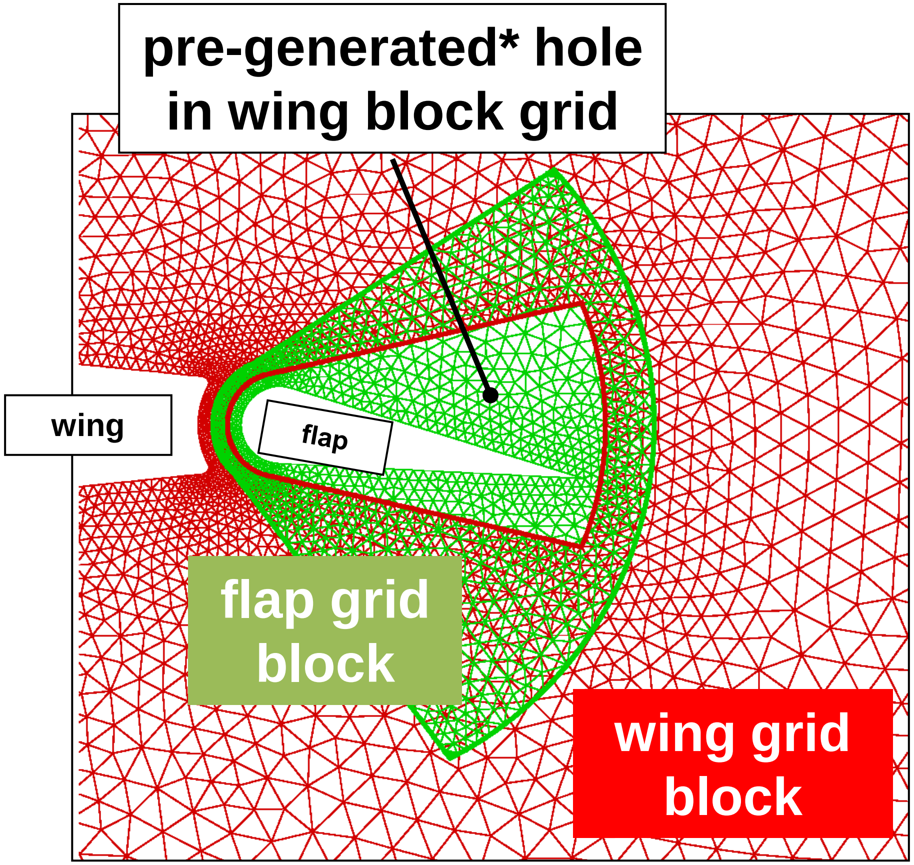
Semi-automatic hole-cutting



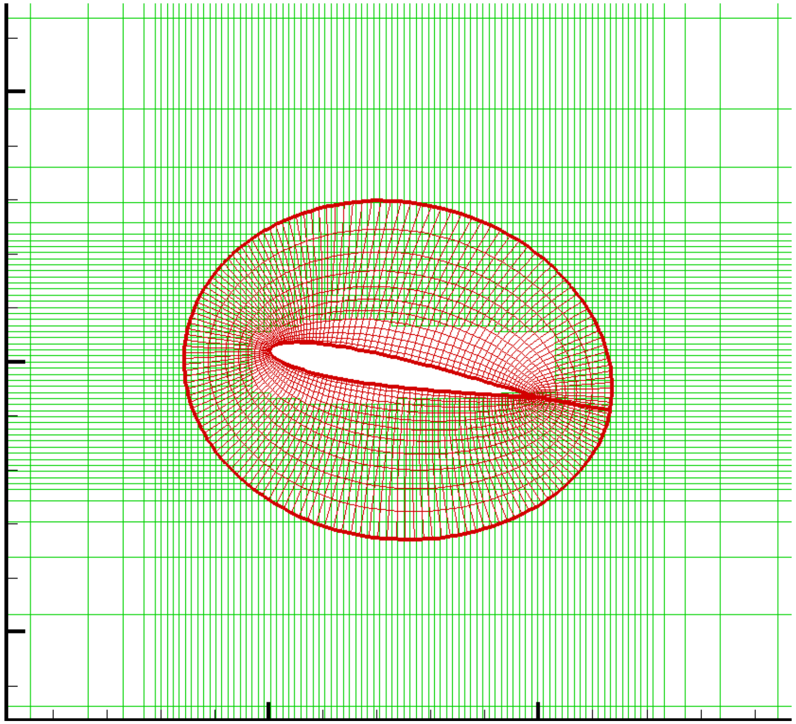
*generated during mesh generation

Our Hole-Cutting Strategies

NO hole-cutting



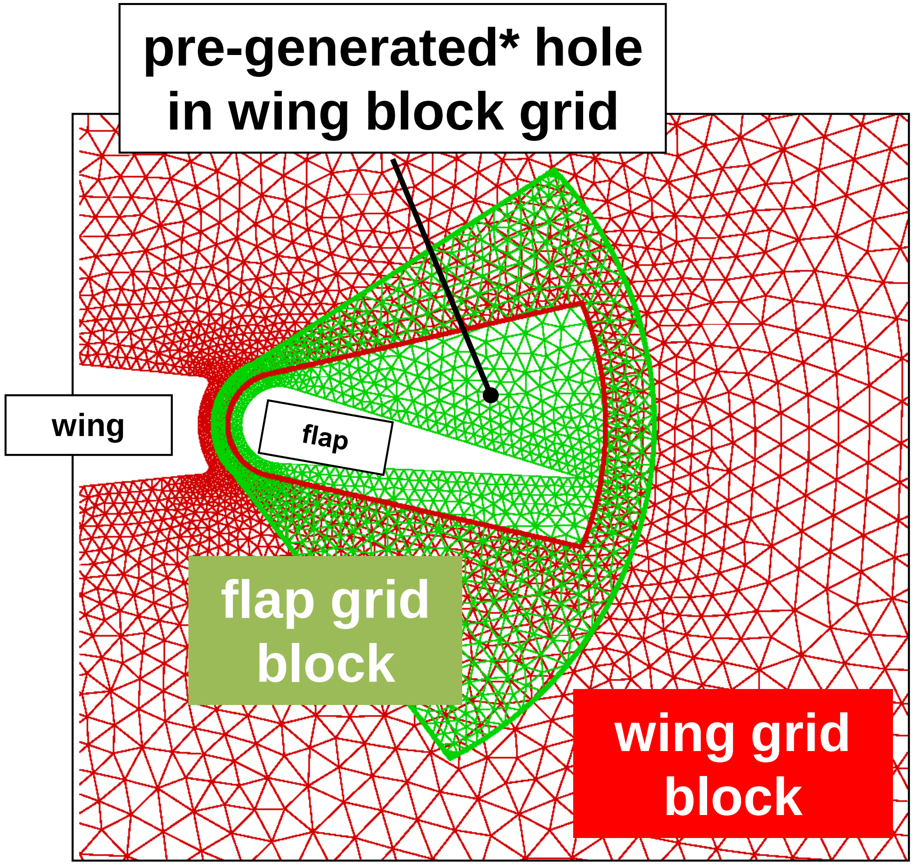
Semi-automatic hole-cutting



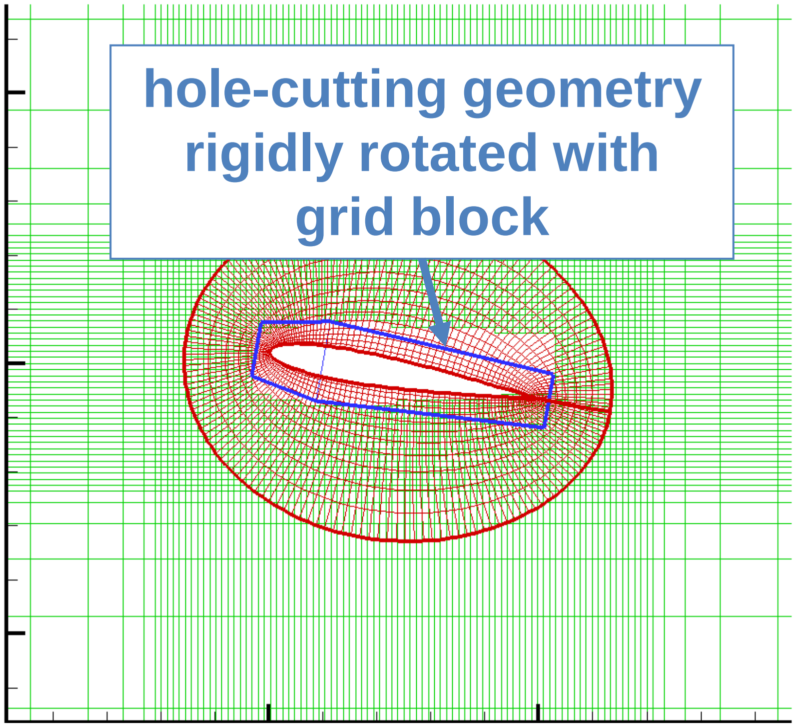
*generated during mesh generation

Our Hole-Cutting Strategies

NO hole-cutting



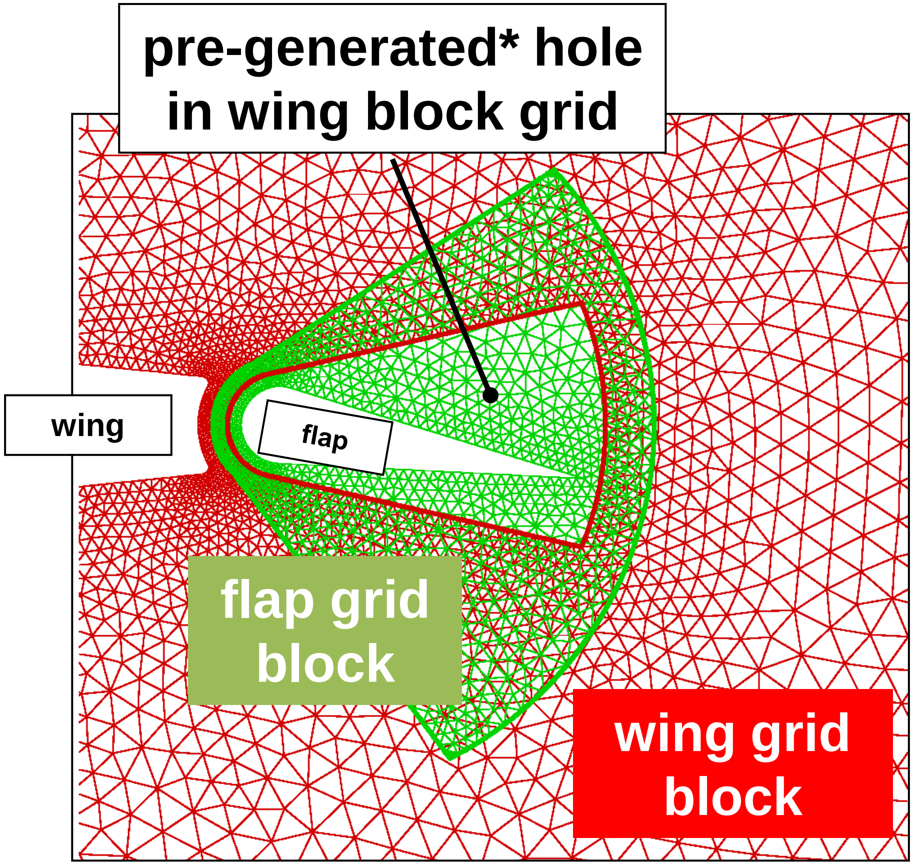
Semi-automatic hole-cutting



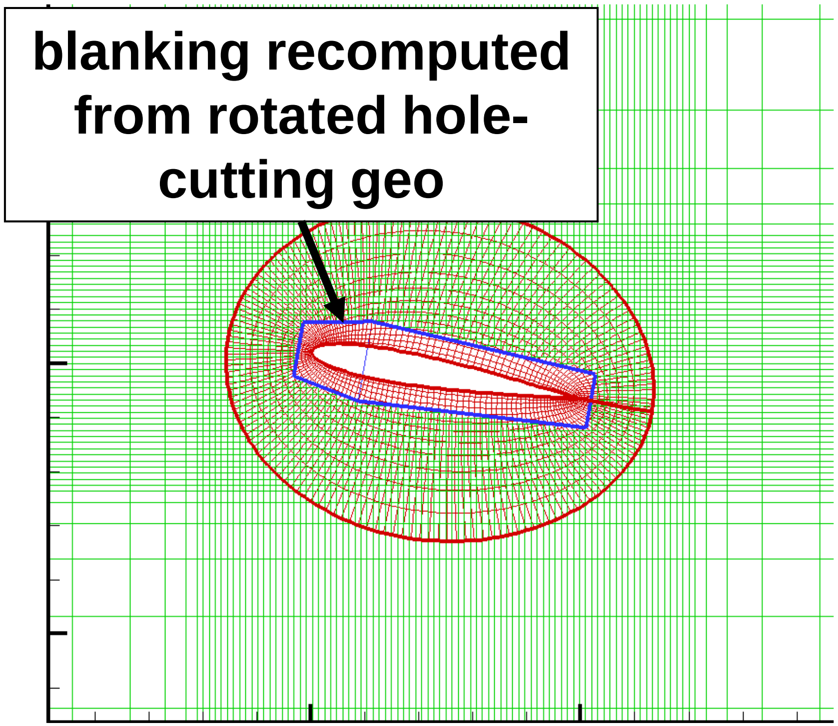
*generated during mesh generation

Our Hole-Cutting Strategies

NO hole-cutting

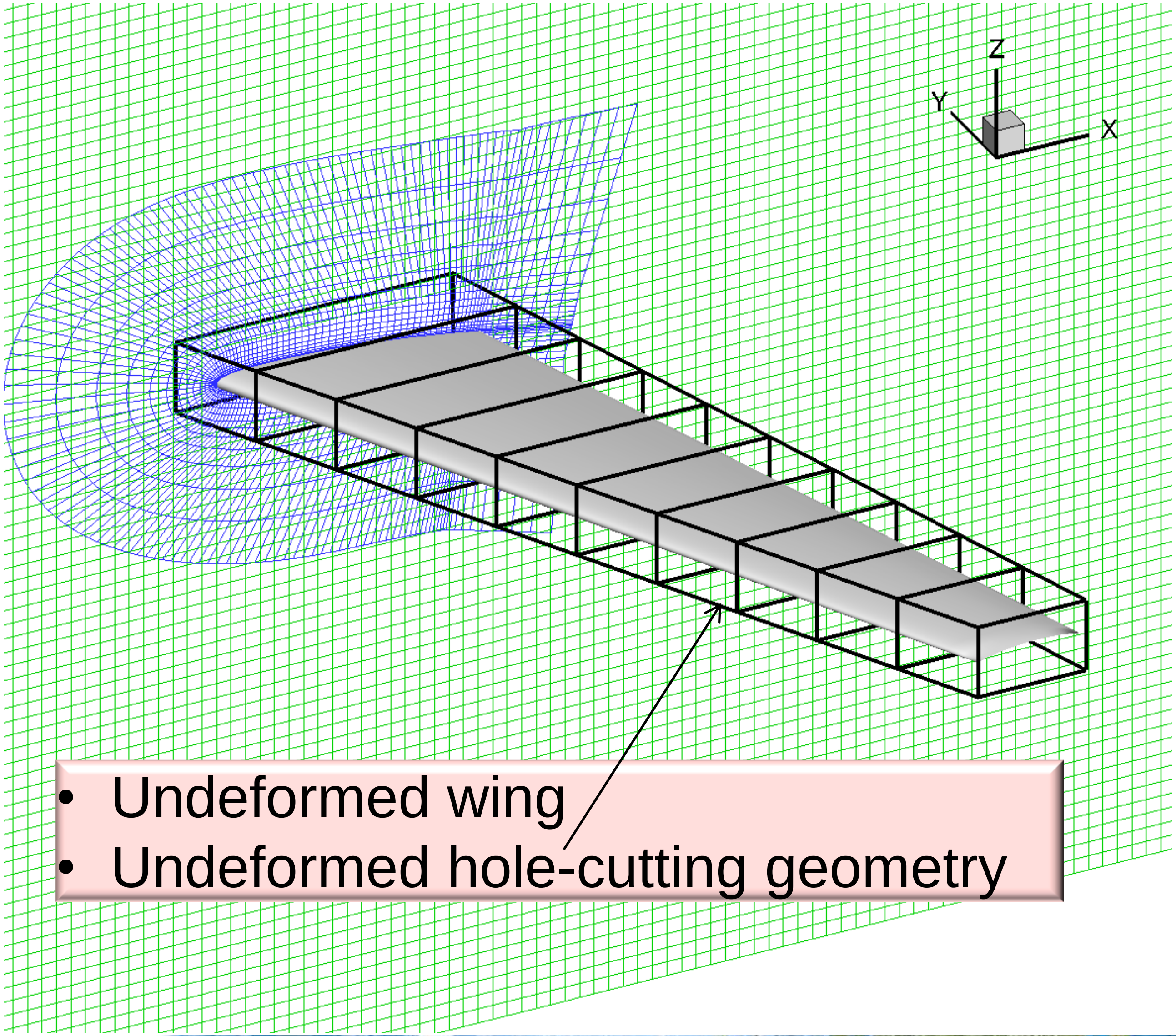


Semi-automatic hole-cutting



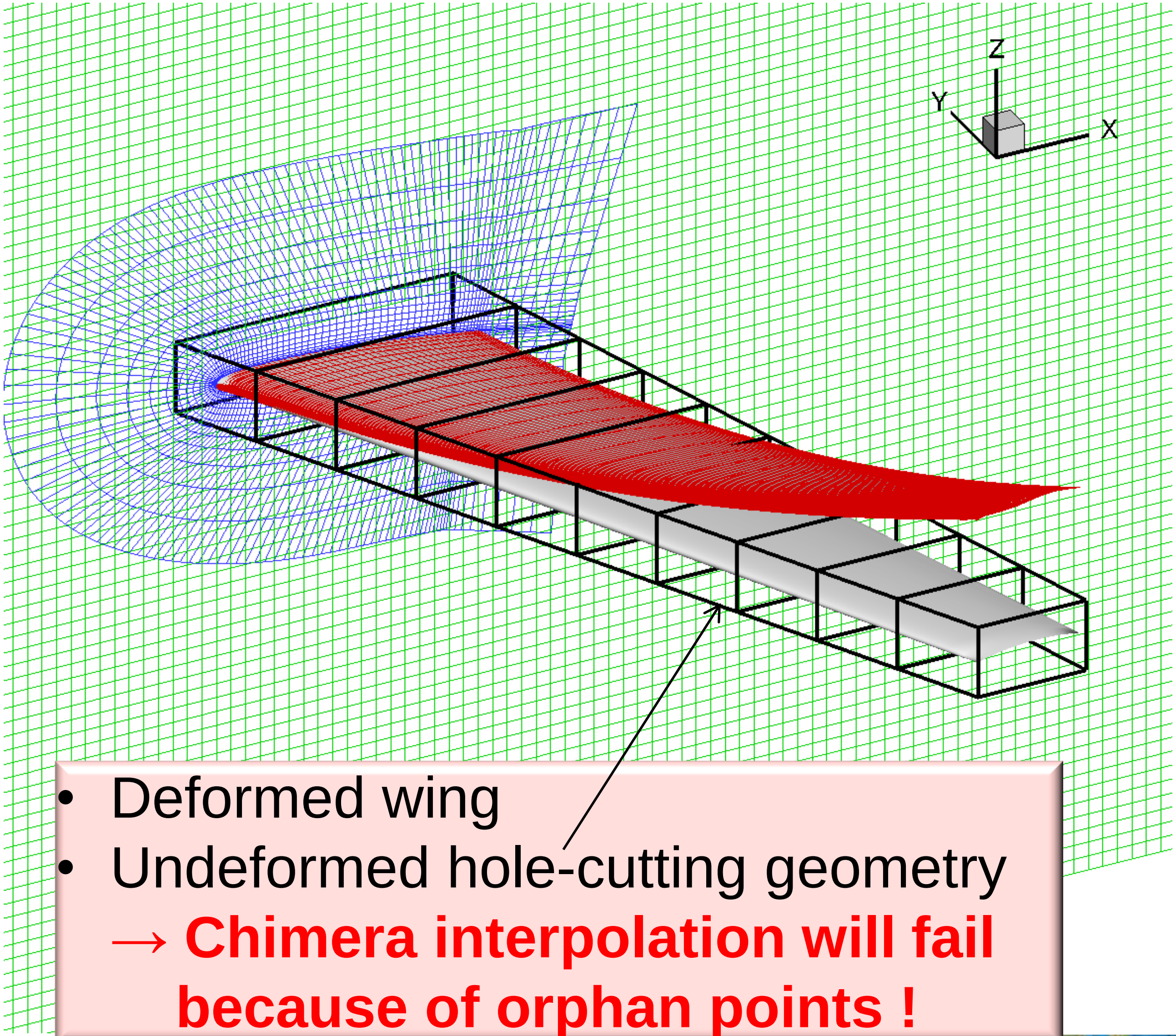
*generated during mesh generation

Hole-Cutting Geometries & Mesh Deformation ?



- Undeformed wing
- Undeformed hole-cutting geometry

Hole-Cutting Geometries & Mesh Deformation ?

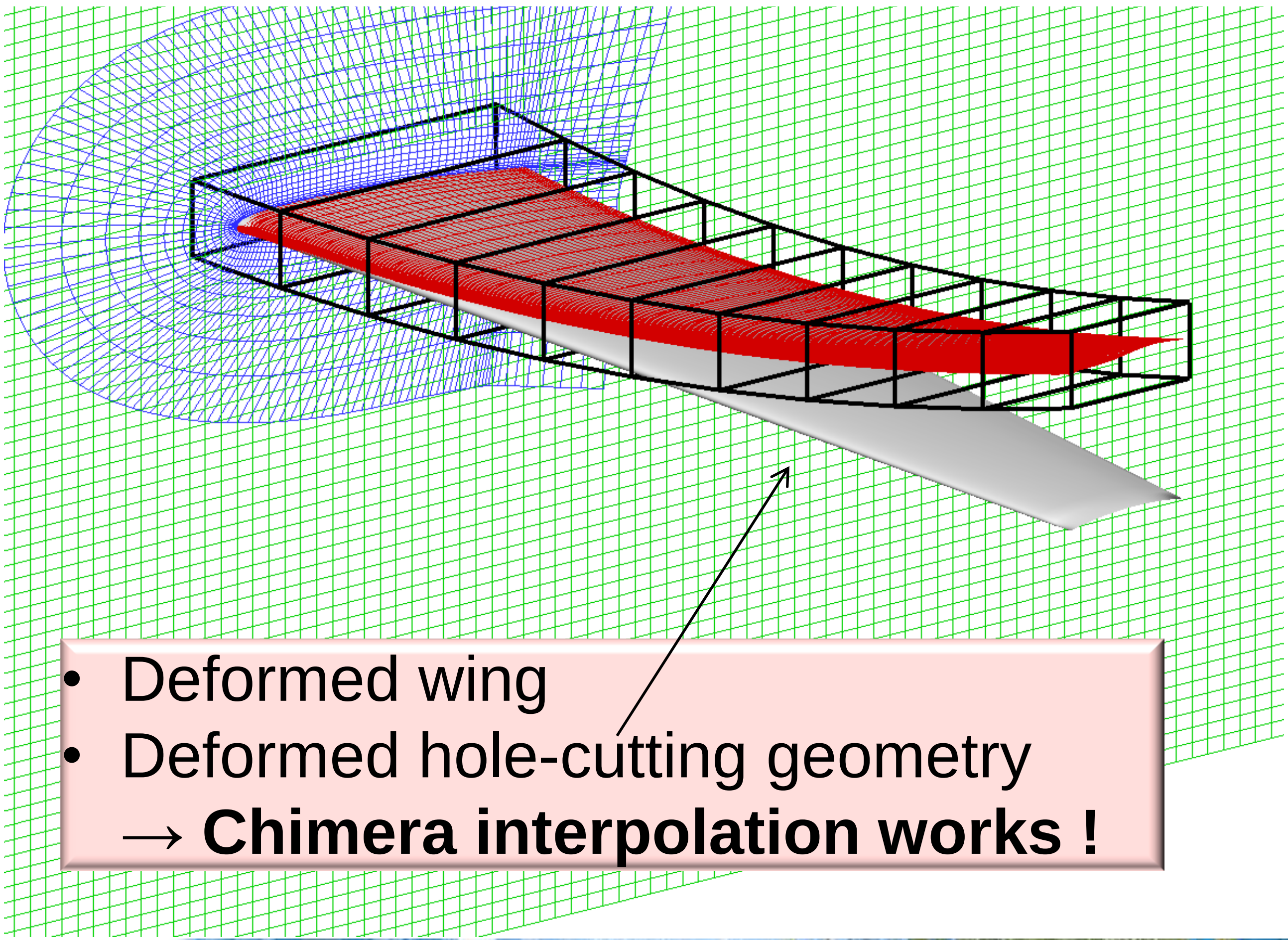


- Deformed wing
- Undeformed hole-cutting geometry

→ Chimera interpolation will fail because of orphan points !

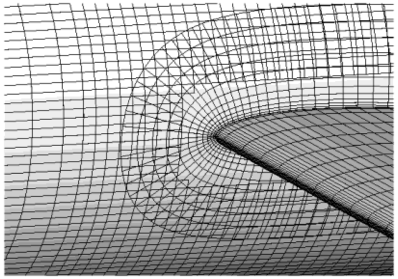
Hole-Cutting Geometries & Mesh Deformation ?

Solution: Mesh deformation operator used for overset grids is also to be applied to hole-cutting geometries

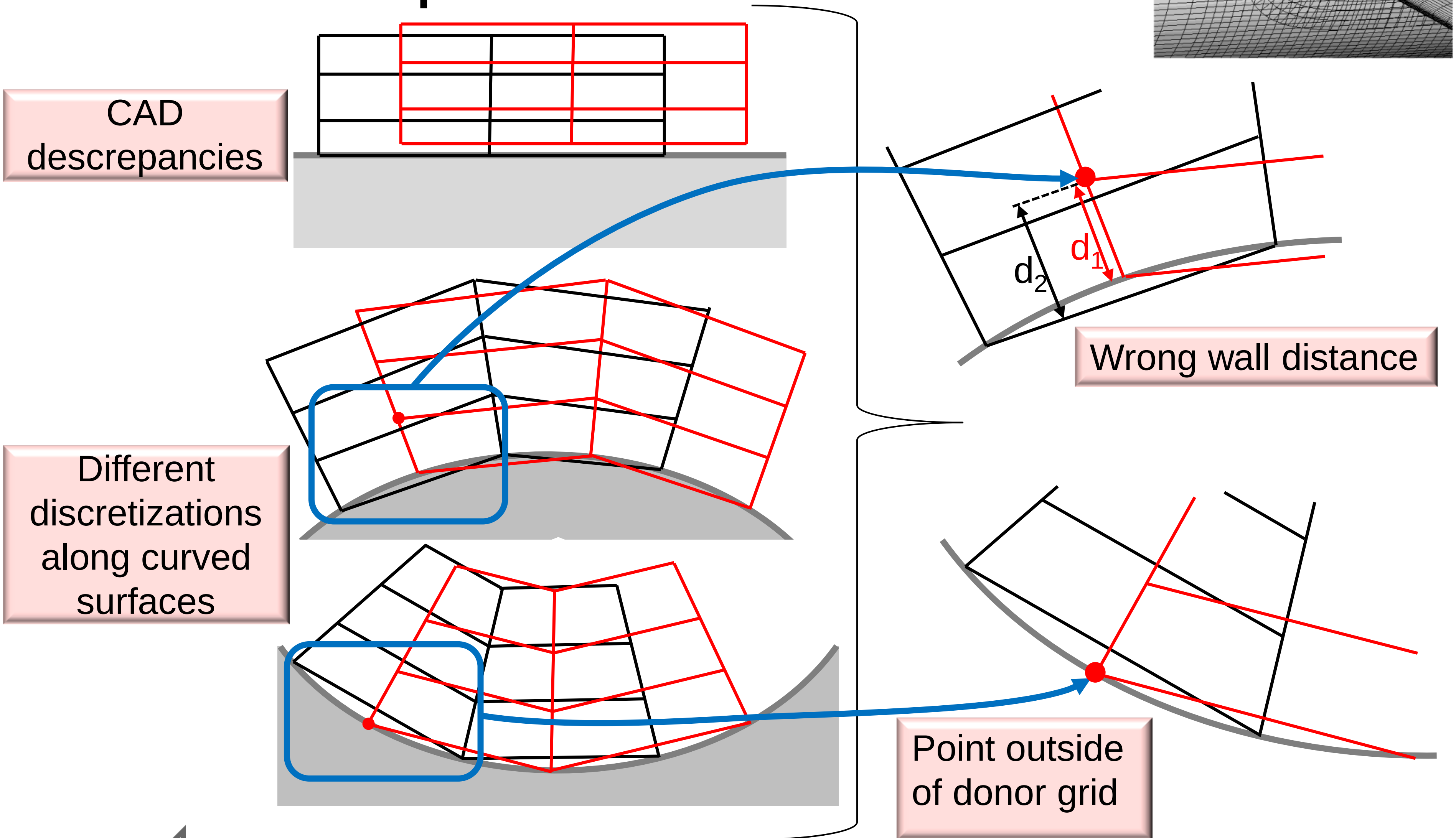


- Deformed wing
- Deformed hole-cutting geometry

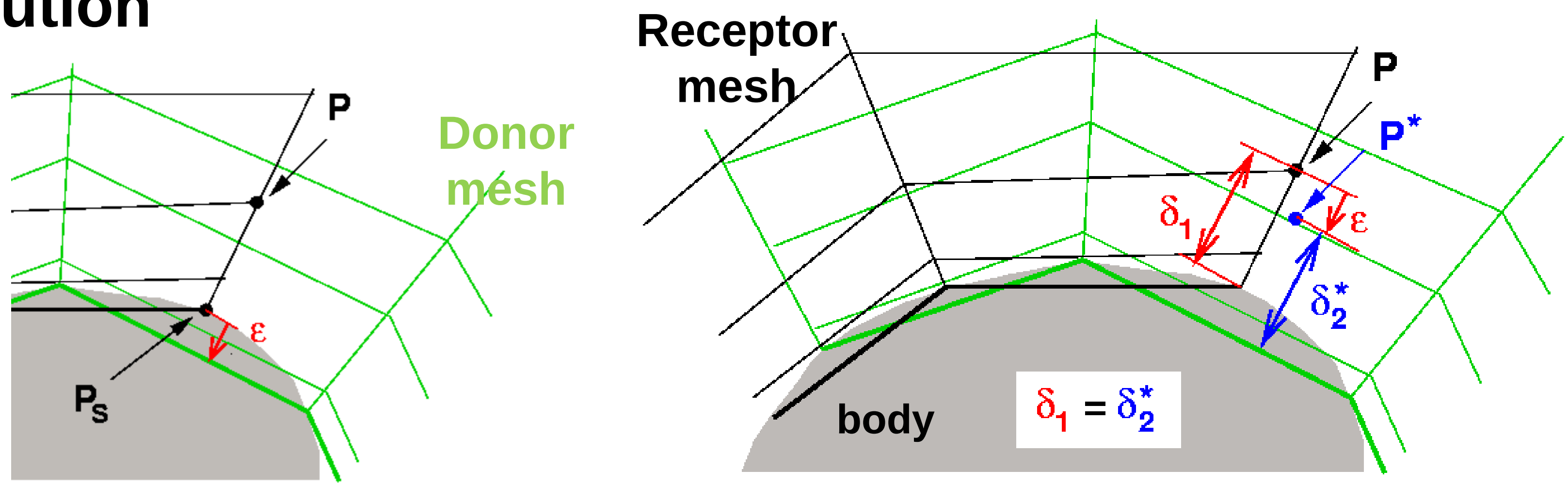
  → **Chimera interpolation works !**



Overlapping Surface Meshes:
Problem Description



Overlapping Surface Meshes:
Solution



A projection method is used, to bypass the problem:

Step 1) Project P to surface of black mesh $\Rightarrow$ point P_S

Step 2) Project Ps in green mesh: $\Rightarrow$ shift vector ε

Step 3) Compute virtual interpolation node P*

$P^* = P + w \cdot \varepsilon$; $w = w(\delta_1)$: Weighting function,

$\delta_1 = 0 : w = 0$; $\delta_1 >> 0 : w = 1$

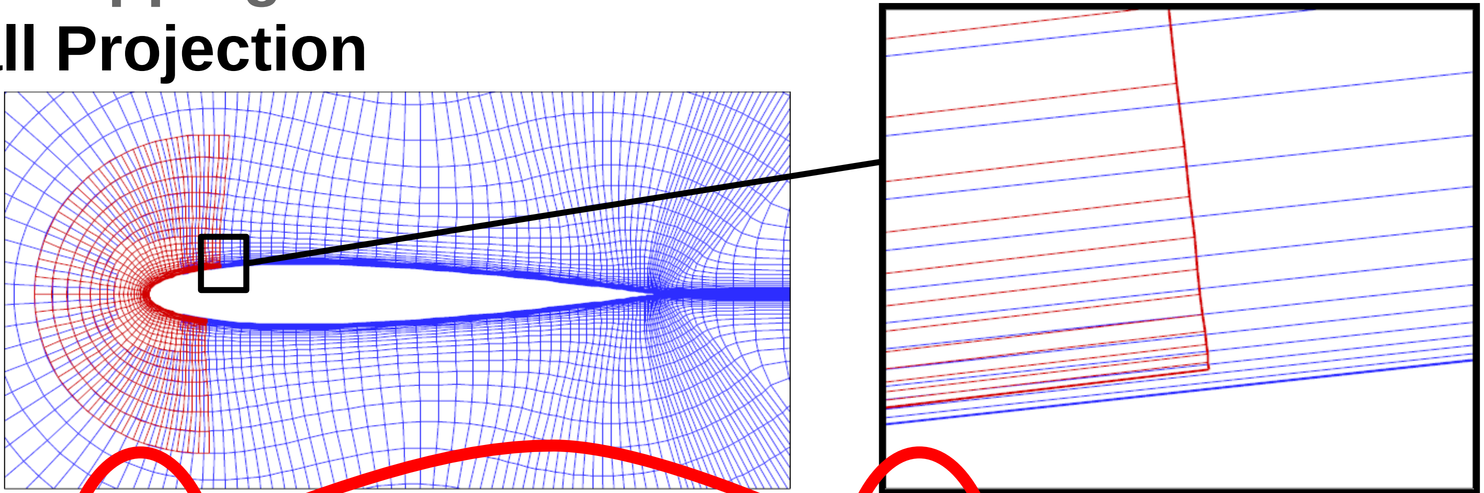
For interpolation nodes P, P* : $\delta_1 = \delta_2^*$ (with $w = 1$)

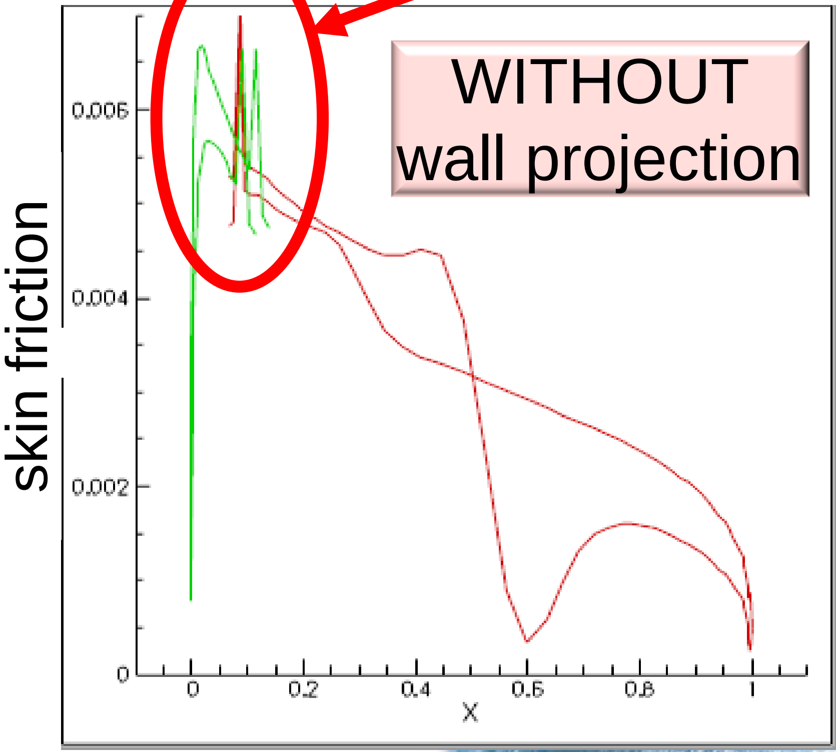
<u>Virtual</u> interpolation node enables accurate interpolation

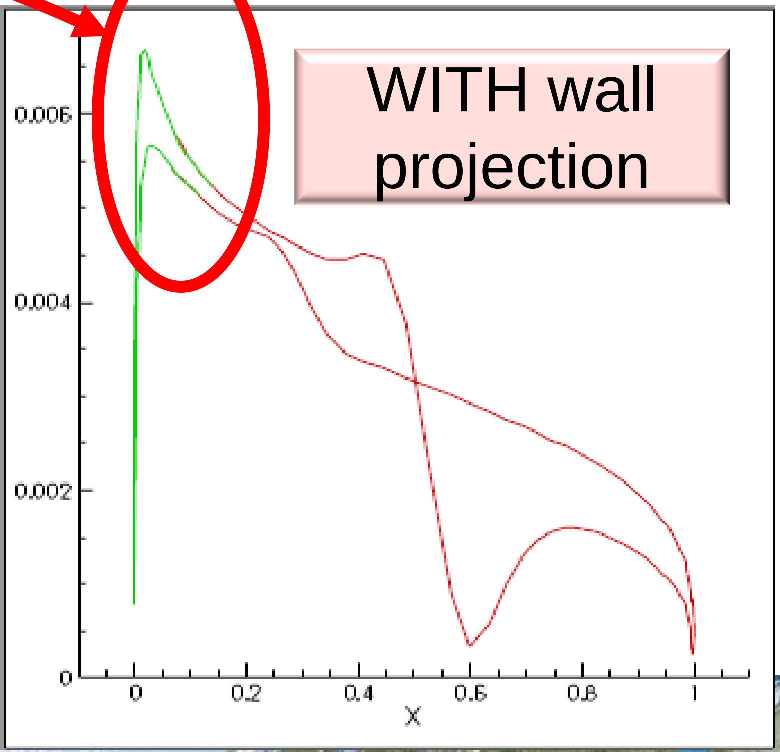
DLR

Overlapping Surface Meshes: Results w/ & w/o Chimera

Wall Projection

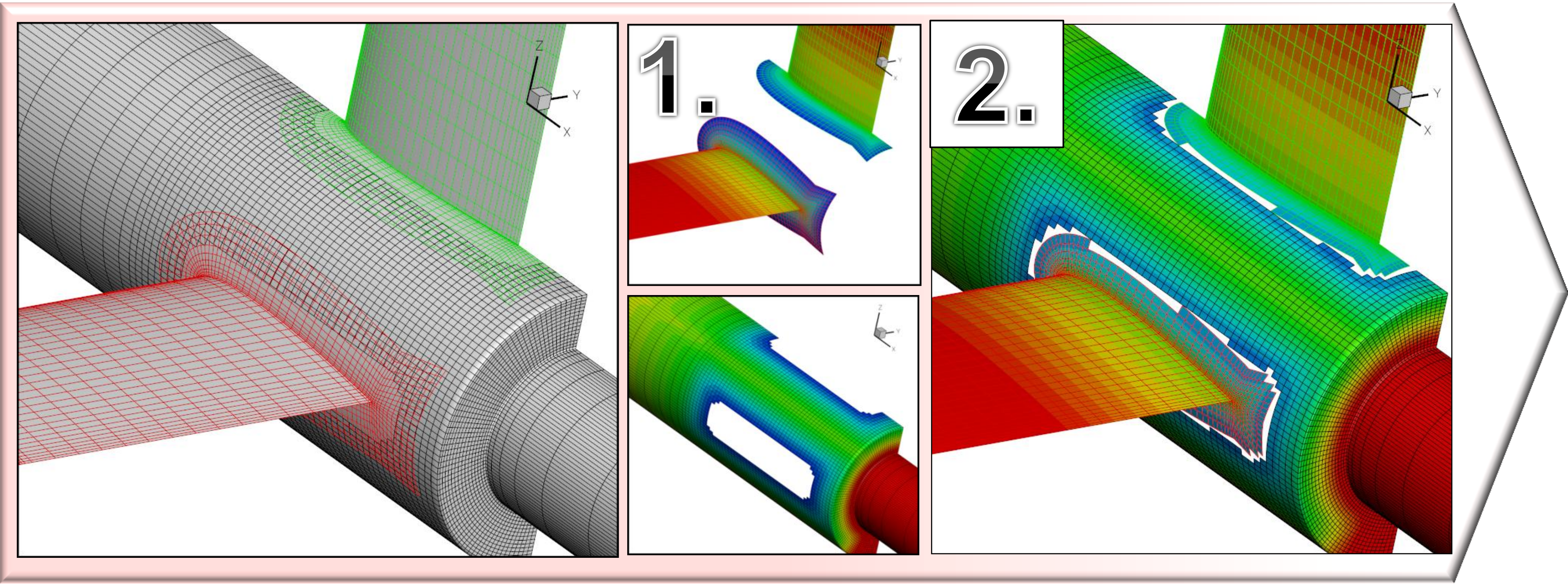






Load Integration on Overlapping Surface Meshes

Problem: Double-counting of cells in regions with overlapping surface grids
=> wrong integral loads



Task: Create an air-tight surface grid , i.e. remove overlap and
fill resulting gaps by triangulation – ***zipper* grids**

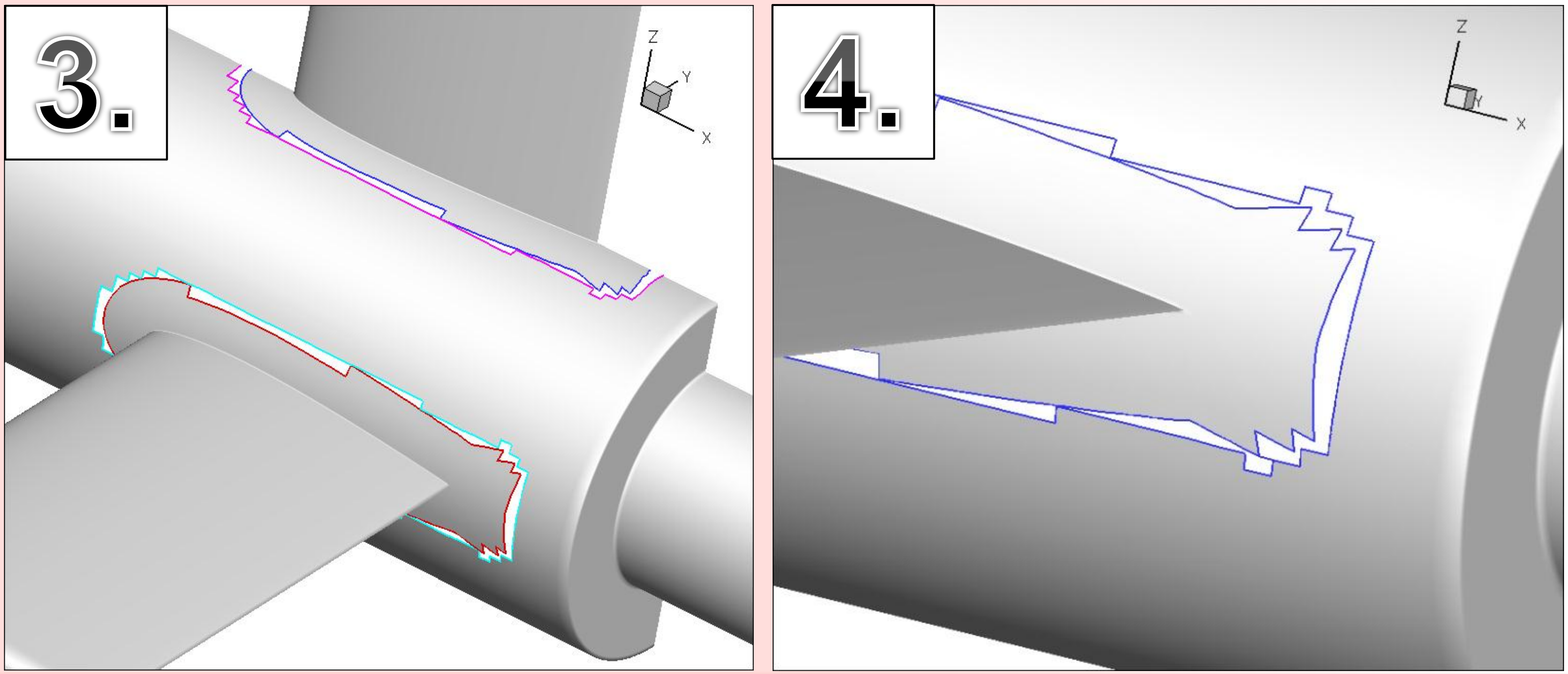
Step 1) Mark cells by distance to Chimera interpolation layer

Step 2) Remove overlapping cells with low priority

=> Overlapping removed , gap has same distance to both interpolation layers

Load Integration on Overlapping Surface Meshes

Problem: Double-counting of cells in regions with overlapping surface grids
=> wrong integral loads



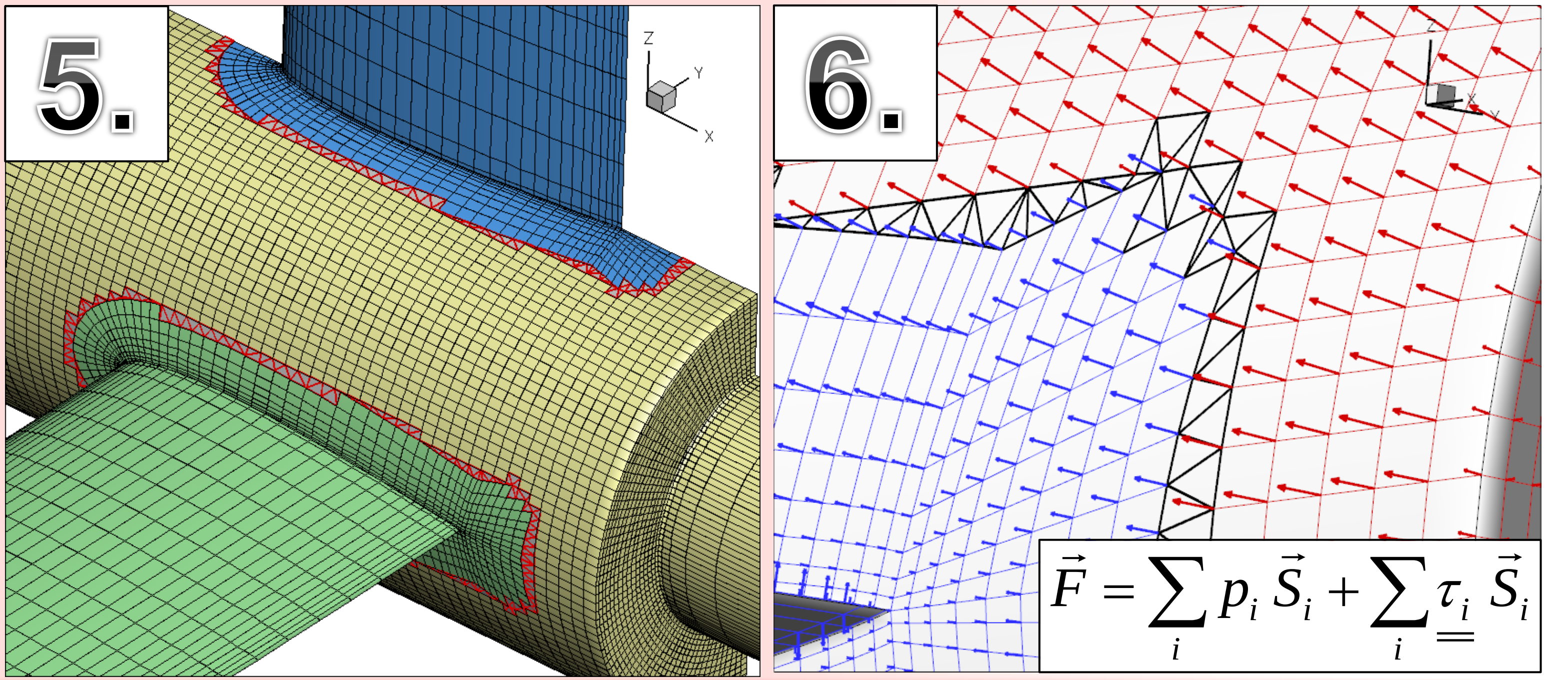
Task: Create a air-tight surface grid = Remove overlap and fill gaps by triangulation – **zipper grids**

Step 3) Extract border lines of resulting grids

Step 4) Gather opposing border lines to build gap data that is to be triangulated

Load Integration on Overlapping Surface Meshes

Problem: Double-counting of cells in regions with overlapping surface grids
=> wrong integral loads



Task: Create a air-tight surface grid = Remove overlap and fill gaps by triangulation – **zipper grids**

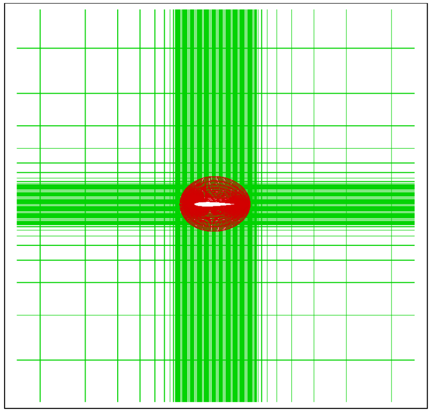
Step 5) Delaunay triangulation

Step 6) Force summation on non-overlapping grid

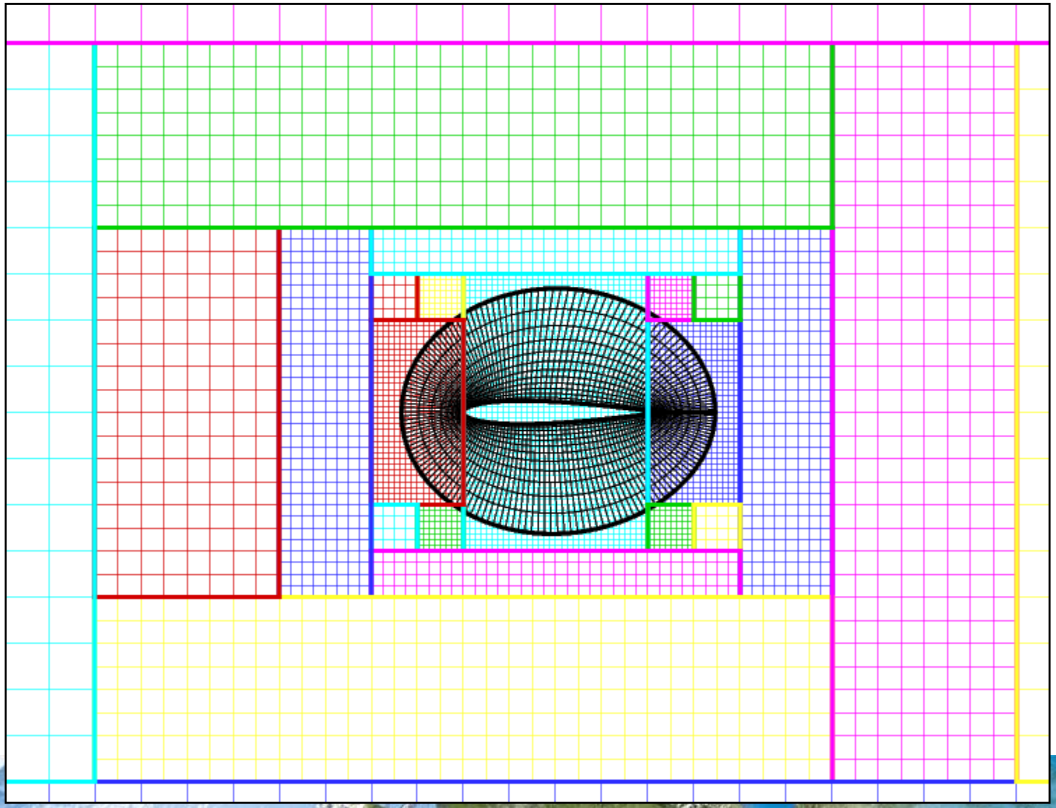
In the past: Only available as sequential postproc. tool
Now: Parallel version directly integrated into TAU

Cartesian Background Grid Generator

Problem: Manual generation of background grid with sufficient overlap & resolution cumbersome and time-consuming



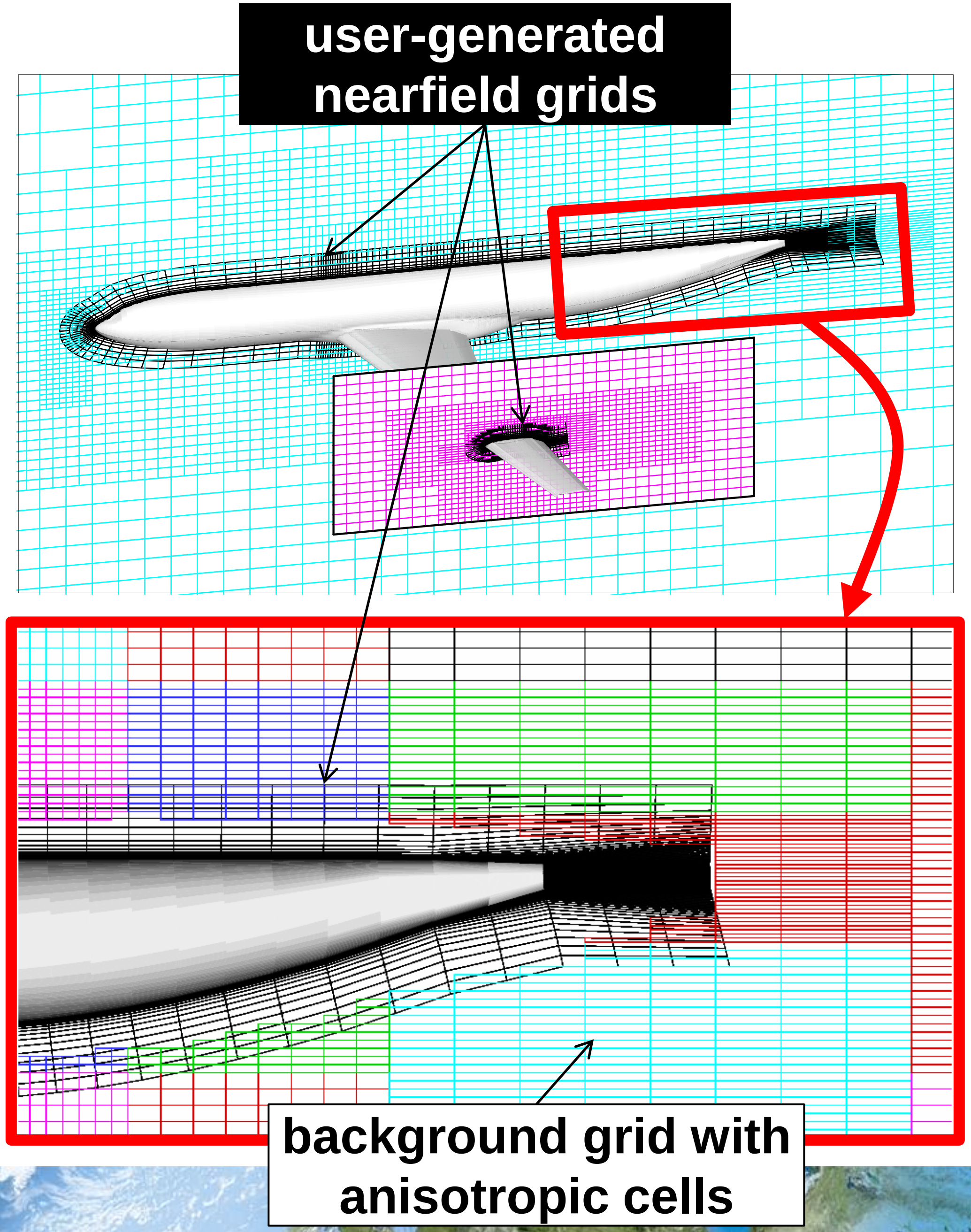
Idea: Automatic generation of Cartesian multiblock grids with hanging nodes (only FLOWer; TAU does not feature hanging nodes capability yet)



Cartesian Background Grid Generator

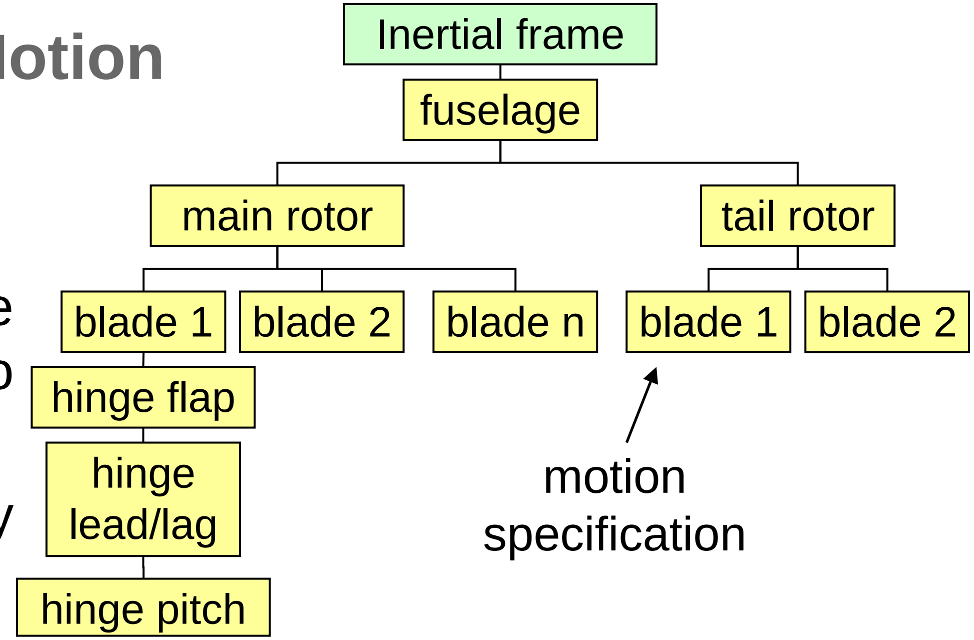
Features:

- Very coarse initial background grids (8x8x8 cells) are adapted consecutively to the nearfield grid's cell sizes
- Anisotropic refinements possible
  => Reduction of interpolation errors
  => Prevents from insufficient cell overlap
- Minimisation of the number of blocks by automatic concatenation



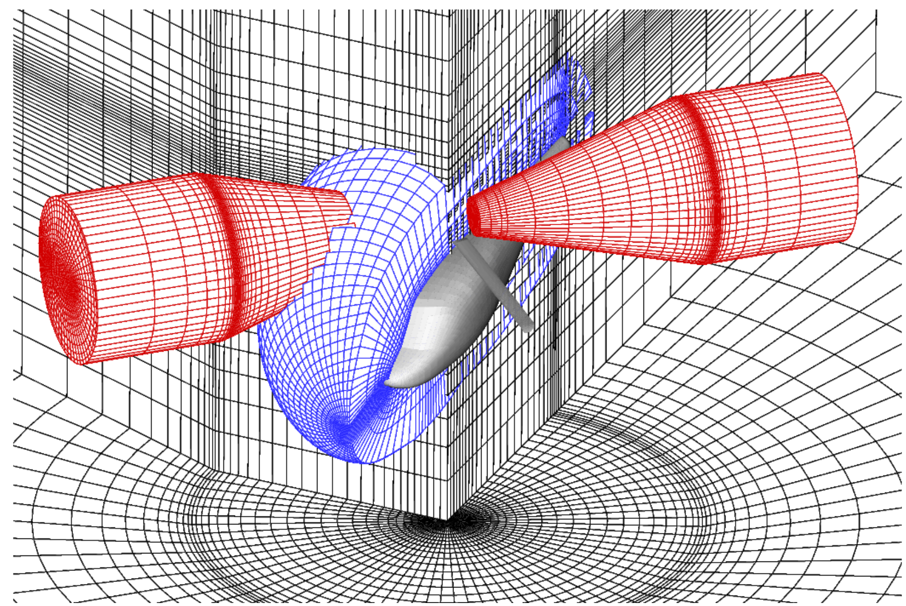
DLR

Overset Grids in Relative Motion



Specification of rigid body motions

- Tree-like data structure (=hierarchy) is used to prescribe grid motion
- Any grid can be linked to any motion specification
- TAU code: grid motion can be set conveniently (required for 6DoF app.) using TAU's Python interfaces

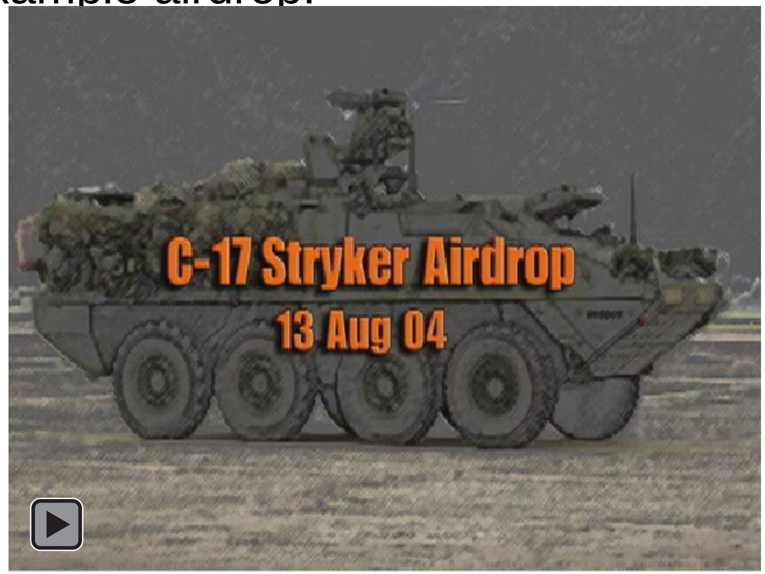


Applications

Coupled CFD-6DoF Simulation of Airdrop Scenario

MiTraPor Test Case

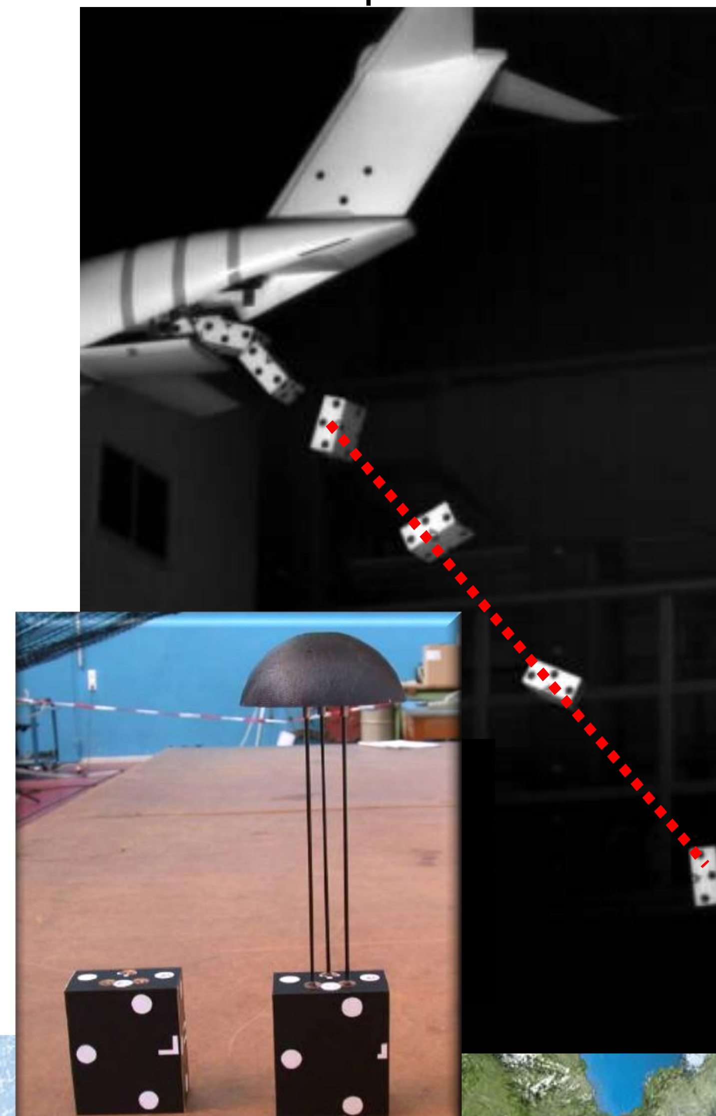
Example airdrop:



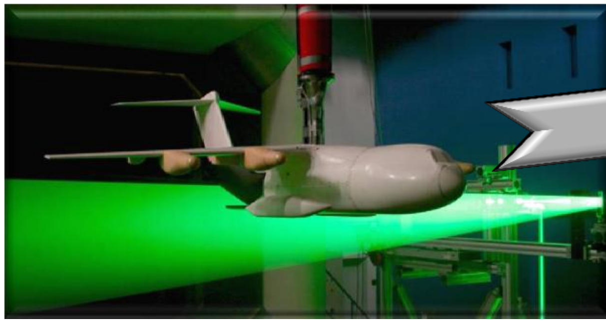
http://www.youtube.com/watch?v=eQwJnVad5L4

- Measurement of location of generic boxes w/ and w/o parachute
- Flow conditions:
  v_{inf} = 18m/s, α = 6°, Re =4x10^5

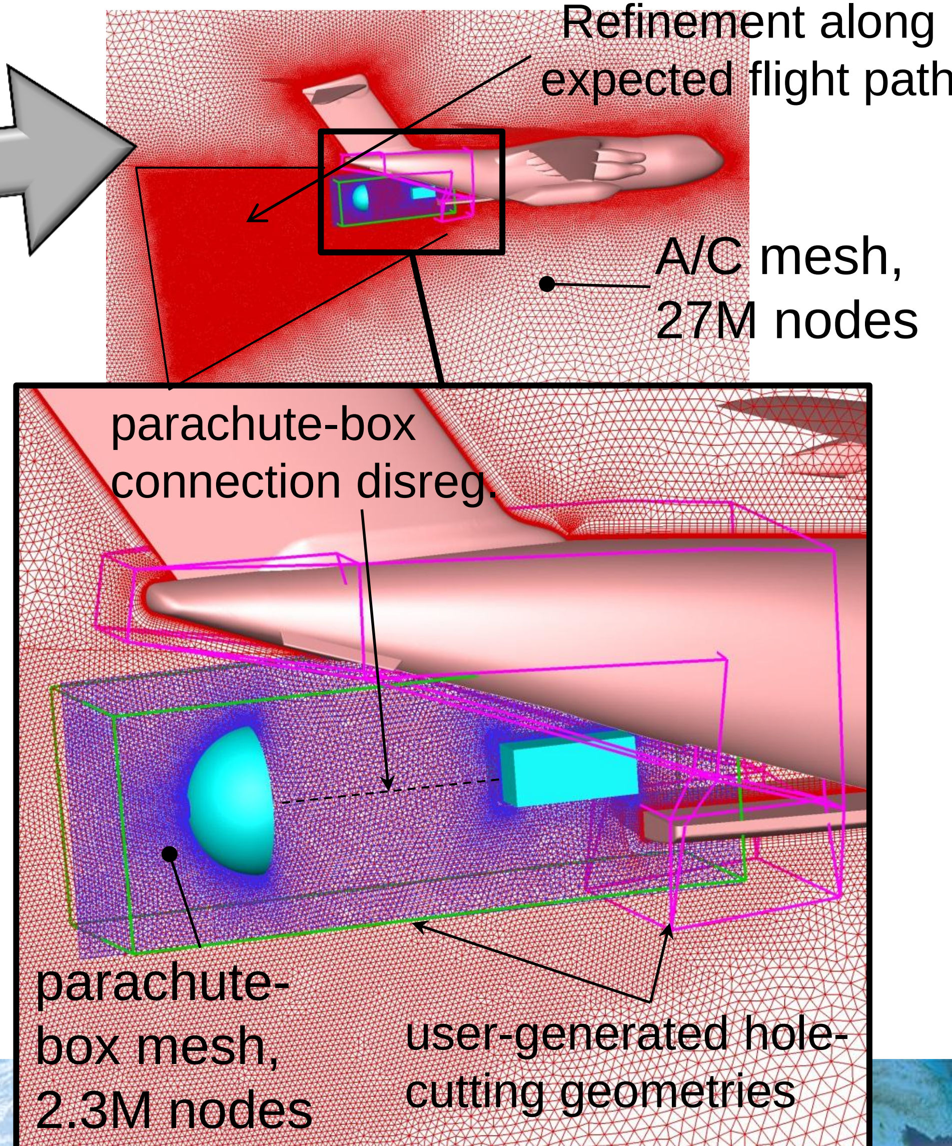
MiTraPor airdrop:



Coupled CFD-6DoF Simulation of Airdrop Scenario
MiTraPor Test Case



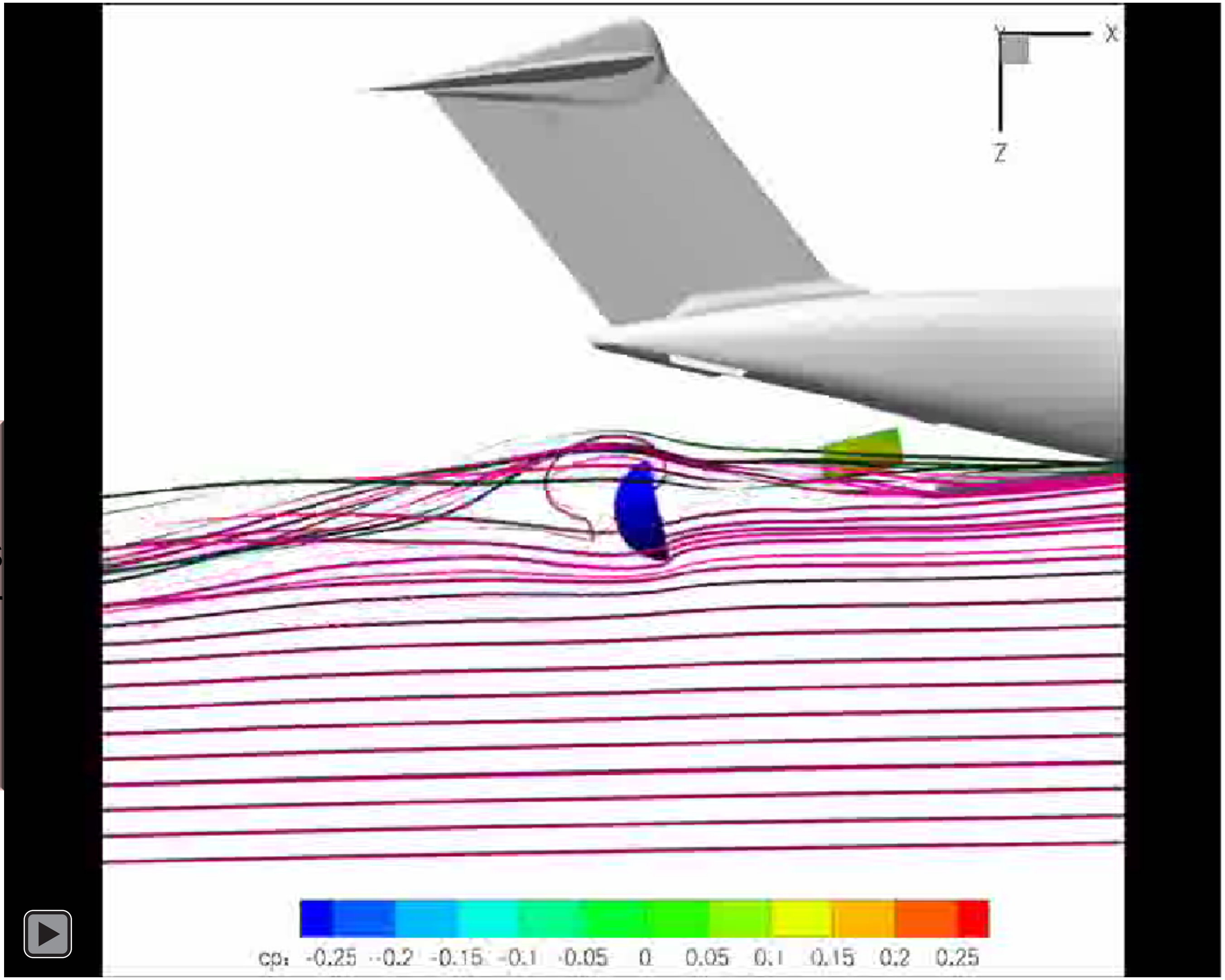
Computational setup:



- Making use of overset grid technique with semi-automatic hole cutting
  - Unstruct. background A/C mesh
  - Unstruct. mesh around parachute-box config.
- Tight coupling of TAU to 6DoF code

Coupled CFD-6DoF Sim. of Airdrop Scenario

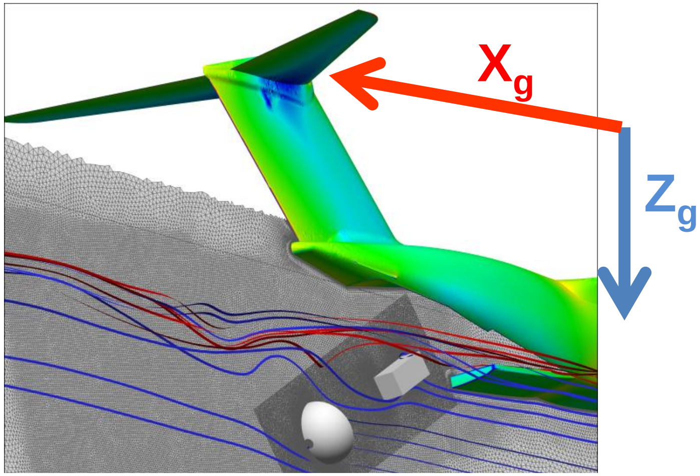
Computing time:
42.6M pts., 144 procs: 160h (~7d) for 0.33s real time, 53min per phys. time step @ 600Hz resol.

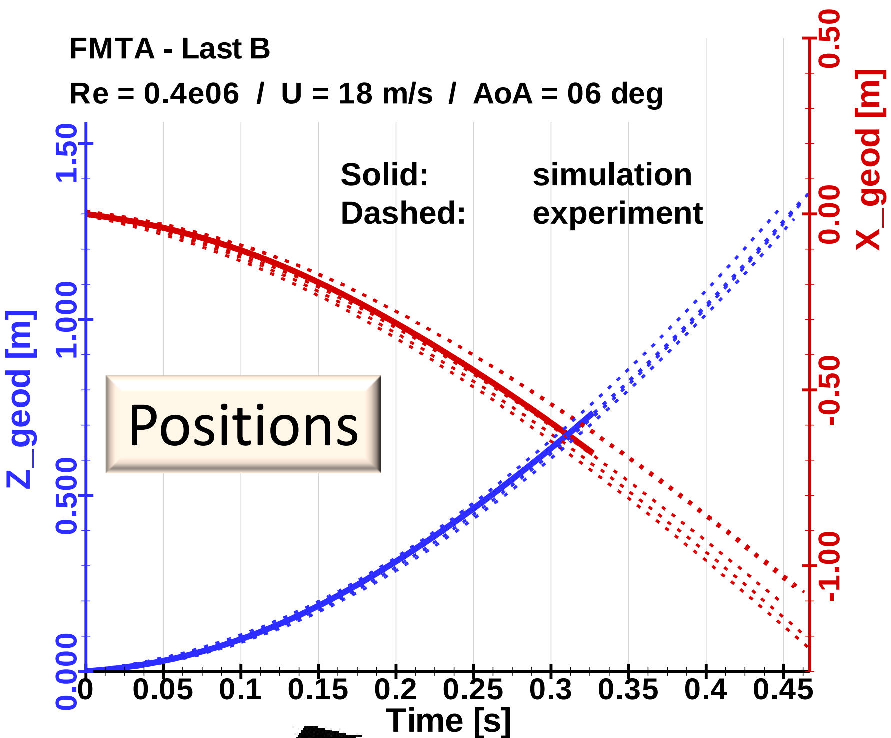


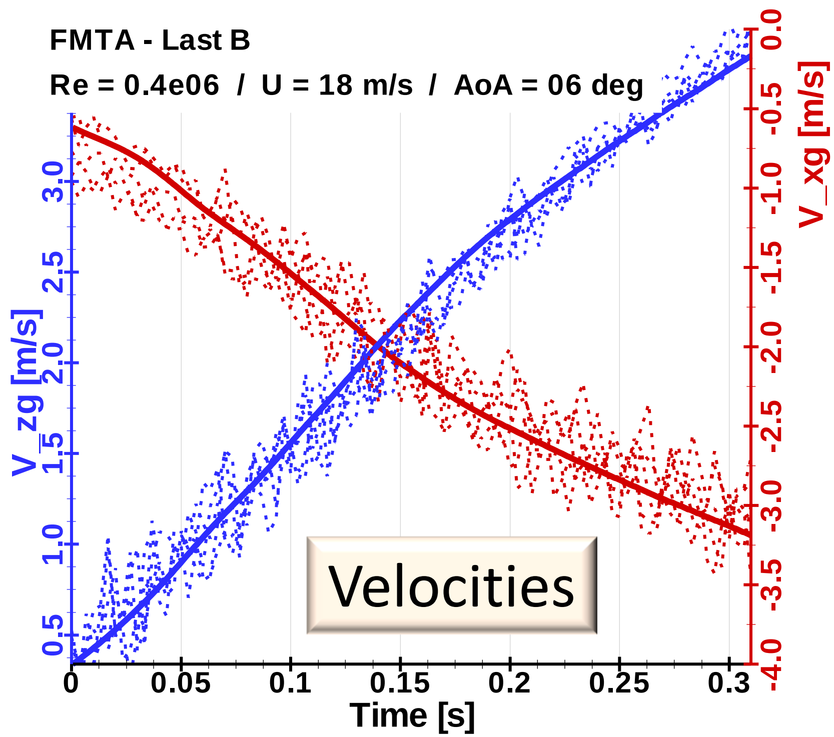
DLR

Coupled CFD-6DoF Simulation of Airdrop Scenario:

MiTraPor Test Case – Comparison of Computed & Measured Trajectories

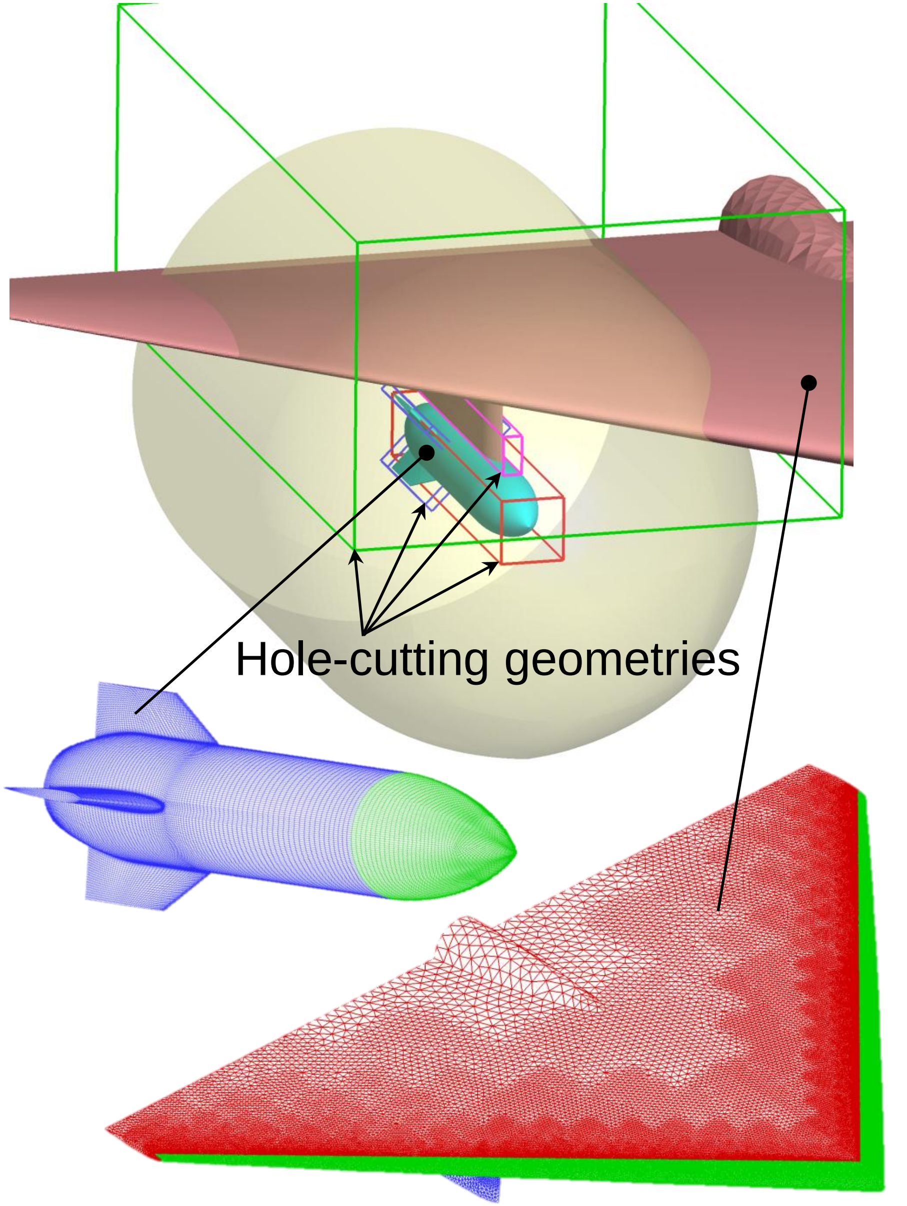






Coupled CFD-6DoF Sim. of Store Release
AFRL* Test Case

- Activity within GARTEUR SIG 47
- Flow conditions: Ma = 0.95, α = 0.0°, Re = 4.1m
- Experiment conducted by AFRL, see RTO-TR-26, ch. 23
- Meshes:
  - **Store**: Structured Ansys ICEM-CFD HEXA Navier-Stokes mesh converted to TAU format, 3.1M nodes
  - **Delta wing**: Unstructured CentaurSoft Centaur Euler mesh, 600k nodes



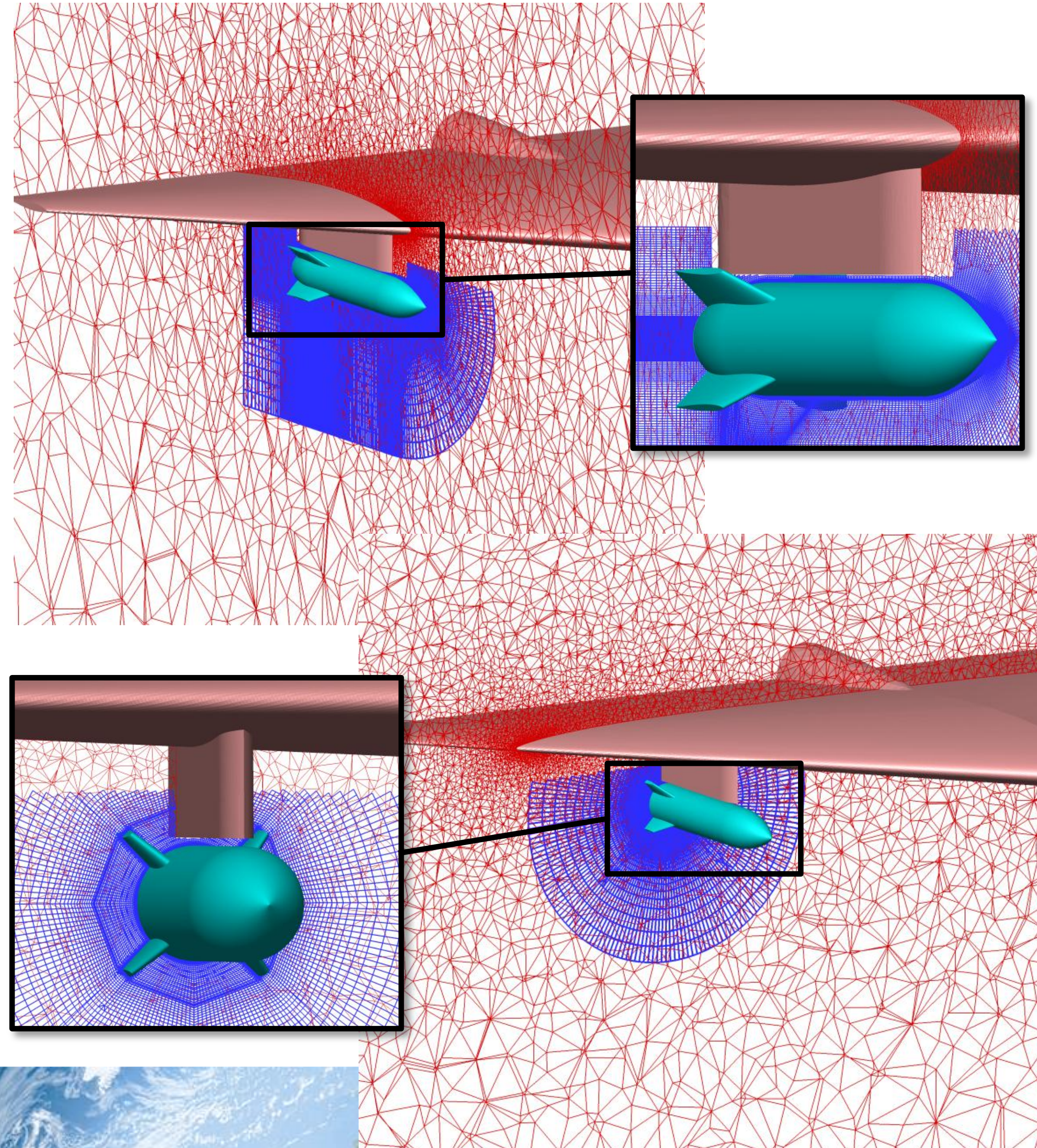
* Air Force Research Laboratory, Wright-Patterson Air Force Base

Coupled CFD-6DoF Sim. of Store Release

AFRL Test Case

Difficulty:

At t=0 store & pylon are very close



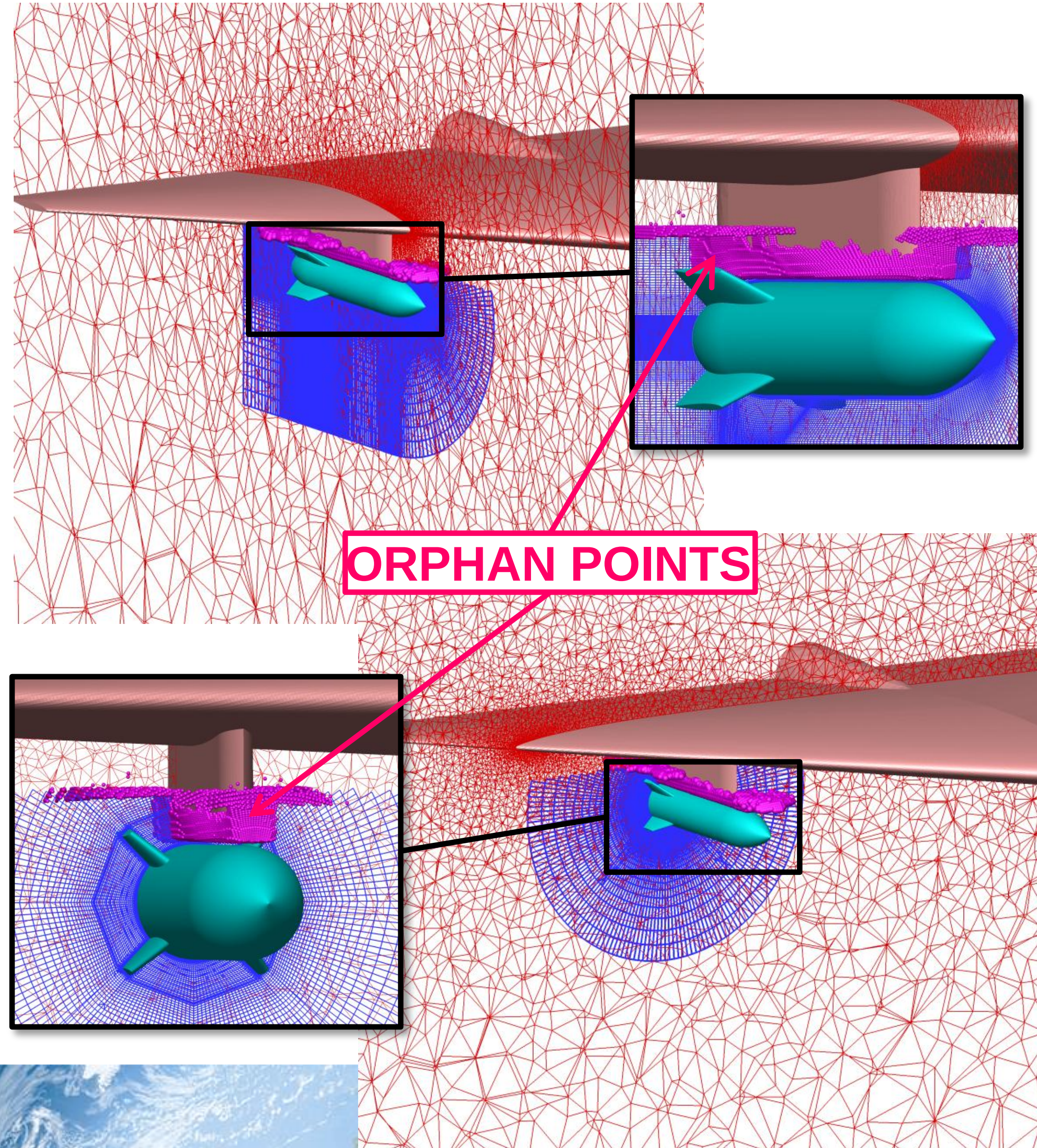
DLR

Coupled CFD-6DoF Sim. of Store Release

AFRL Test Case

Difficulty:

At t=0 store & pylon are very close => With the original mesh, not all donor cells could be found



Coupled CFD-6DoF Sim. of Store Release

AFRL Test Case

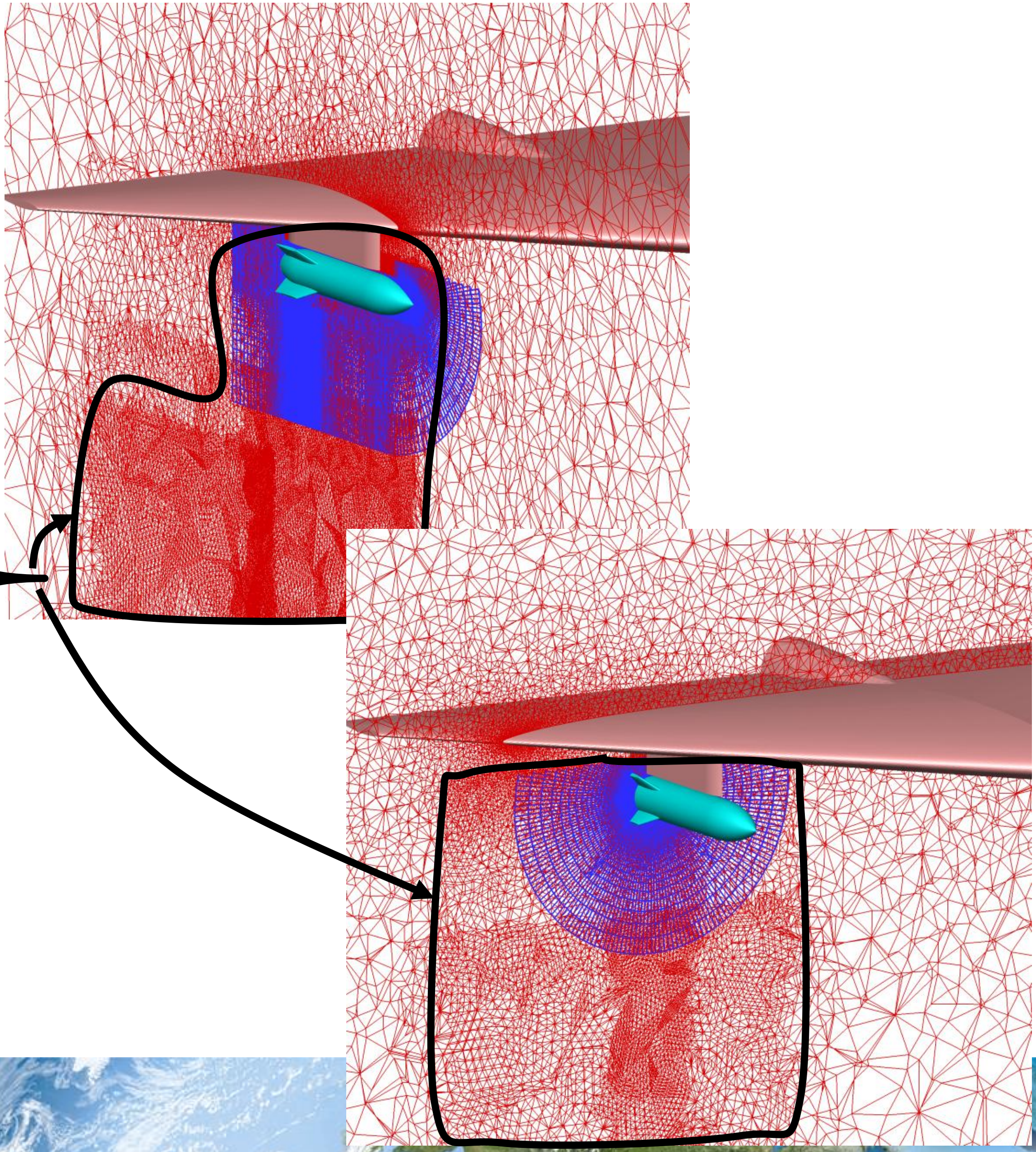
Difficulty:

At t=0 store & pylon are very close => With the original mesh, not all donor cells could be found

Approach for solving the problem:

Uniform adaptation of the delta wing mesh

- … close to the pylon
- … in expected flight path area

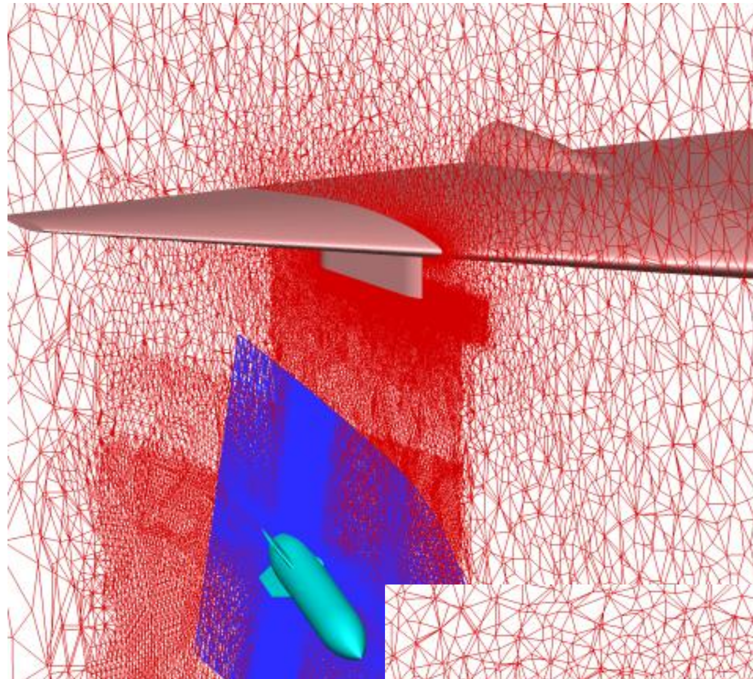


Coupled CFD-6DoF Sim. of Store Release

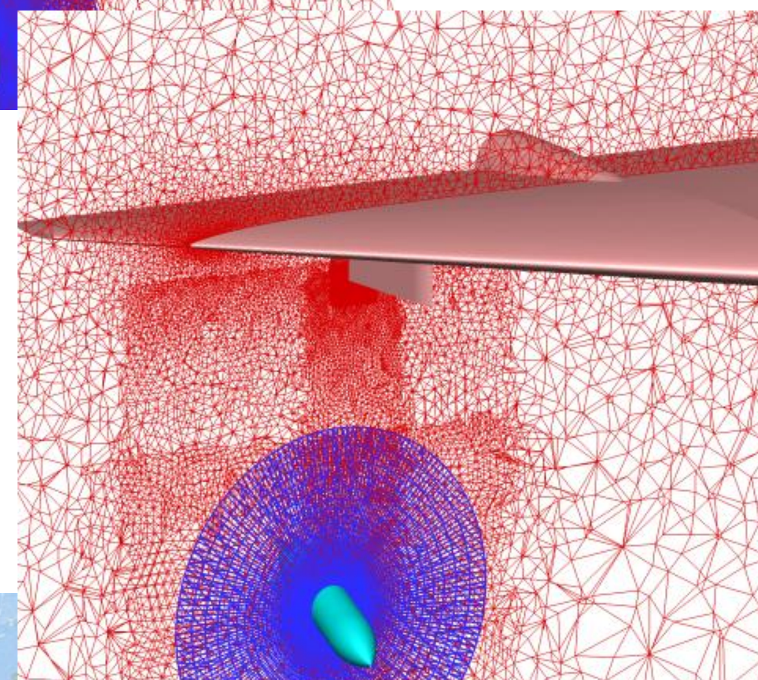
AFRL Test Case

Difficulty:
At t=0 store & pylon are very close => With the original mesh, not all donor cells could be found

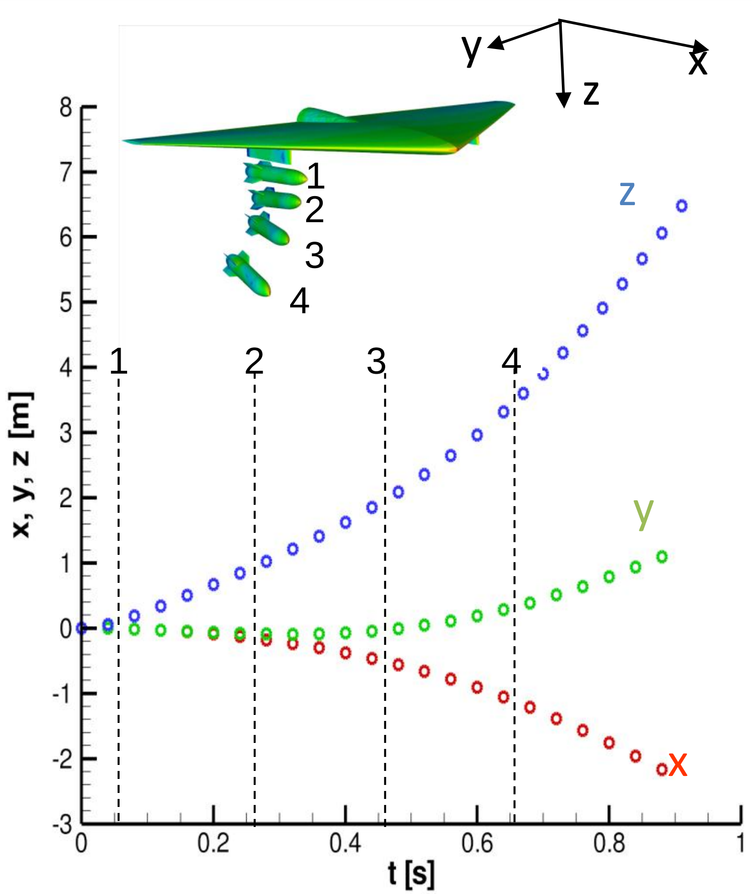
Approach for solving the problem:
Uniform adaptation of the delta wing mesh
- … close to the pylon
- … in expected flight path area



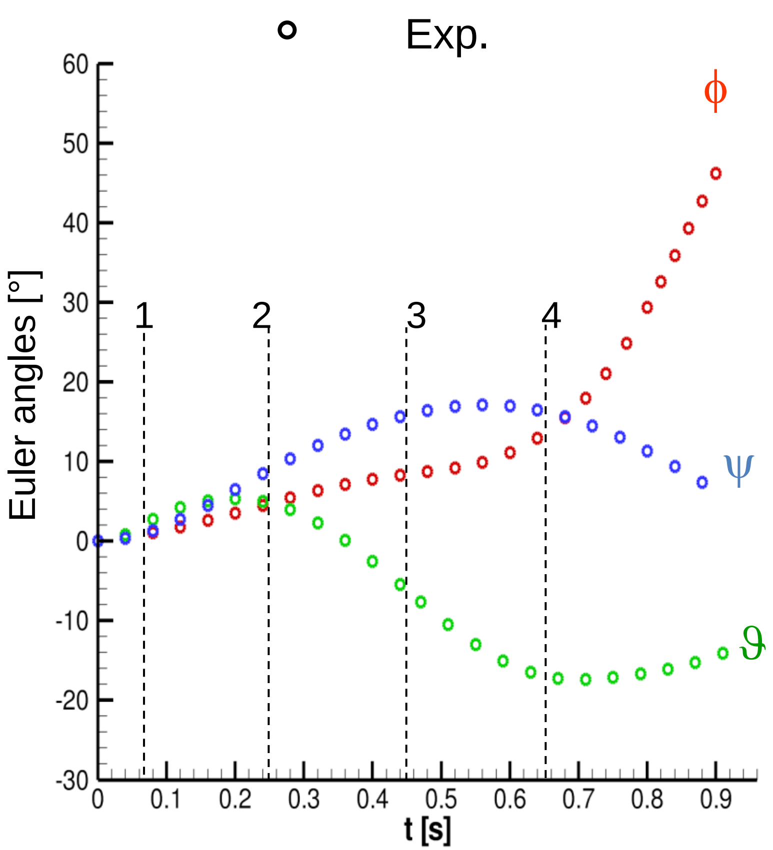
⇒With adaptation approach the simulation worked



Coupled CFD-6DoF Simulation of Store Release
AFRL Test Case



Locations



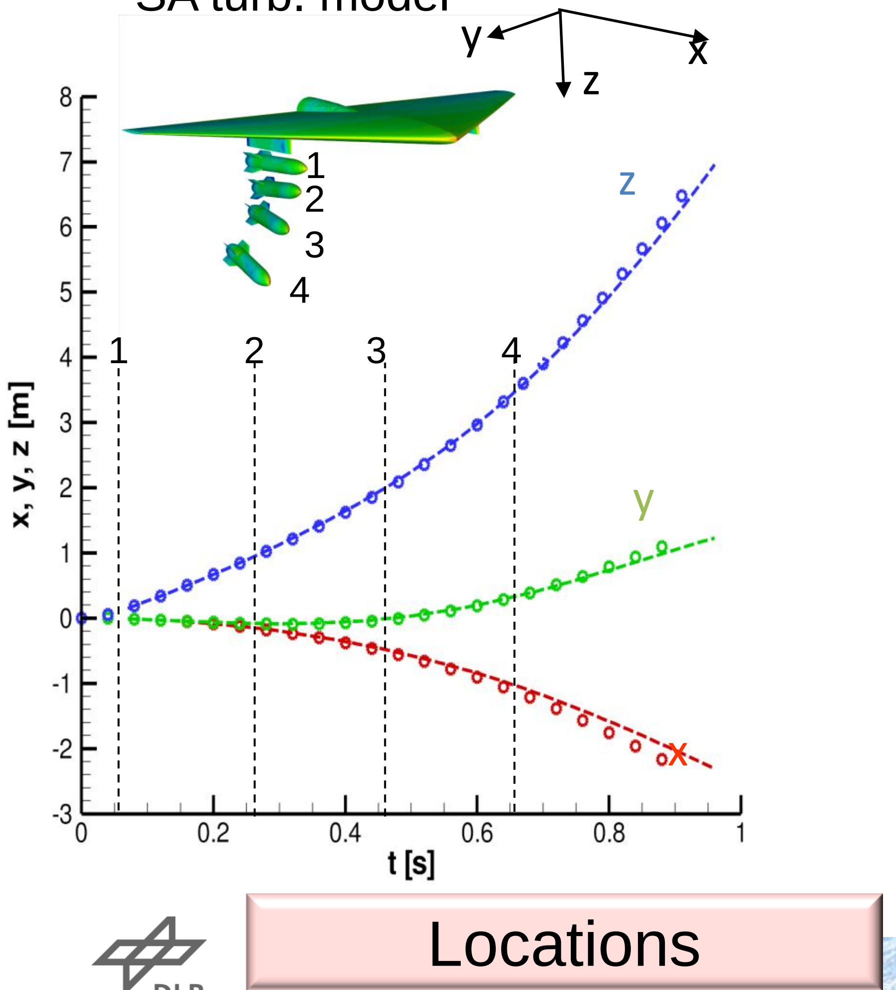
Rotations

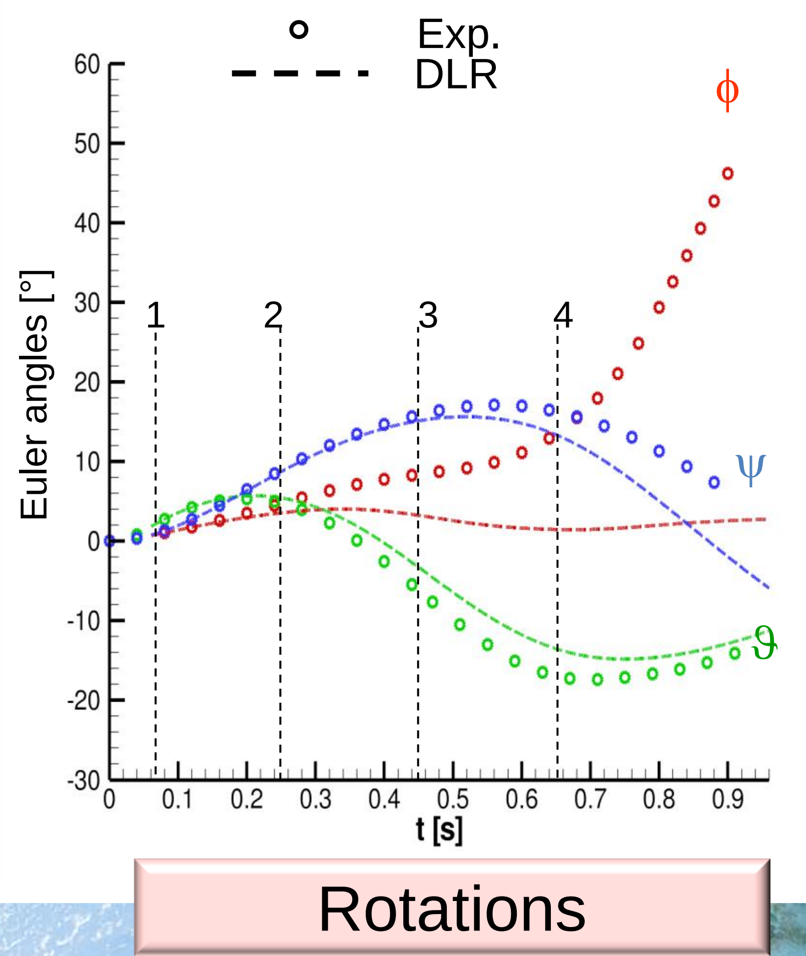
Coupled CFD-6DoF Simulation of Store Release

AFRL Test Case

DLR's simulation:

- Strong coupling of CFD-6DoF (unsteady)
- SA turb. model





Coupled CFD-6DoF Simulation of Store Release

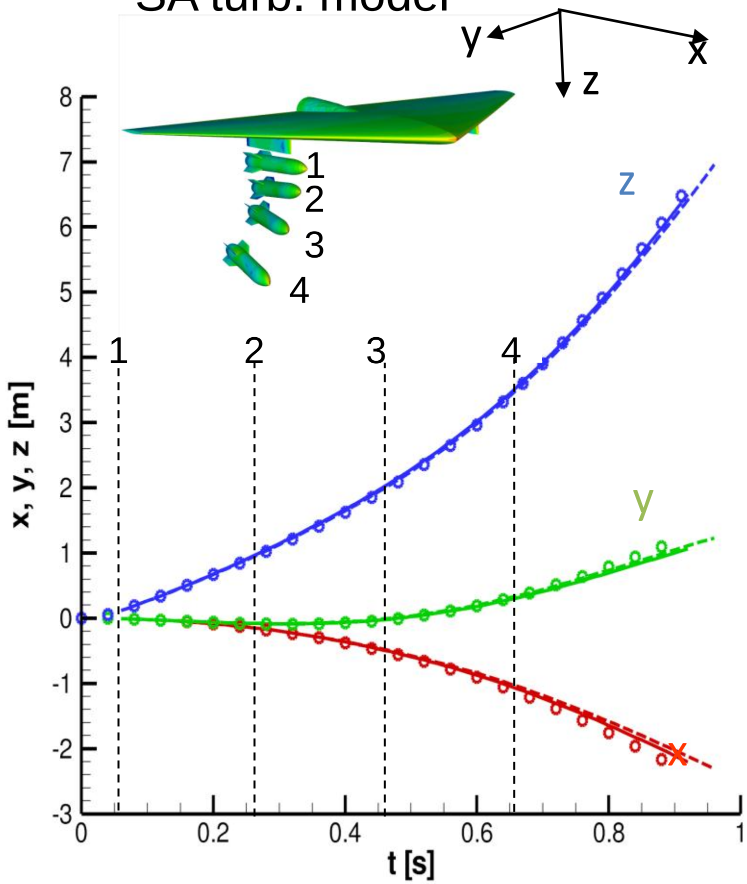
AFRL Test Case

DLR's simulation:

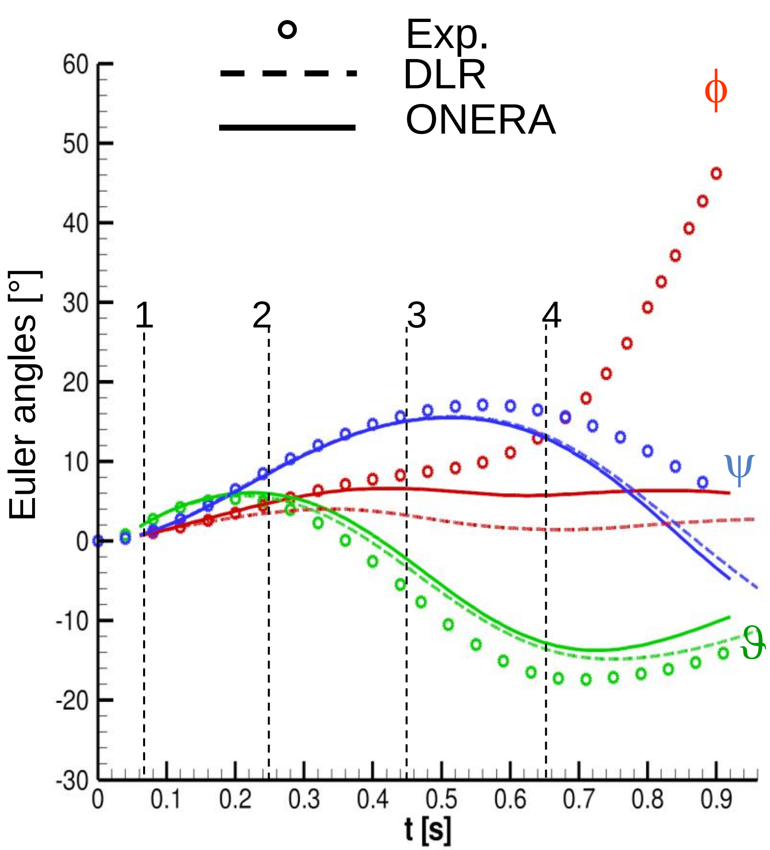
- Strong CFD-6DoF coupling (unsteady)
- SA turb. model

ONERA's simulation:

- Quasi-steady coupling CFD-6DoF
- SA turb. model



Locations



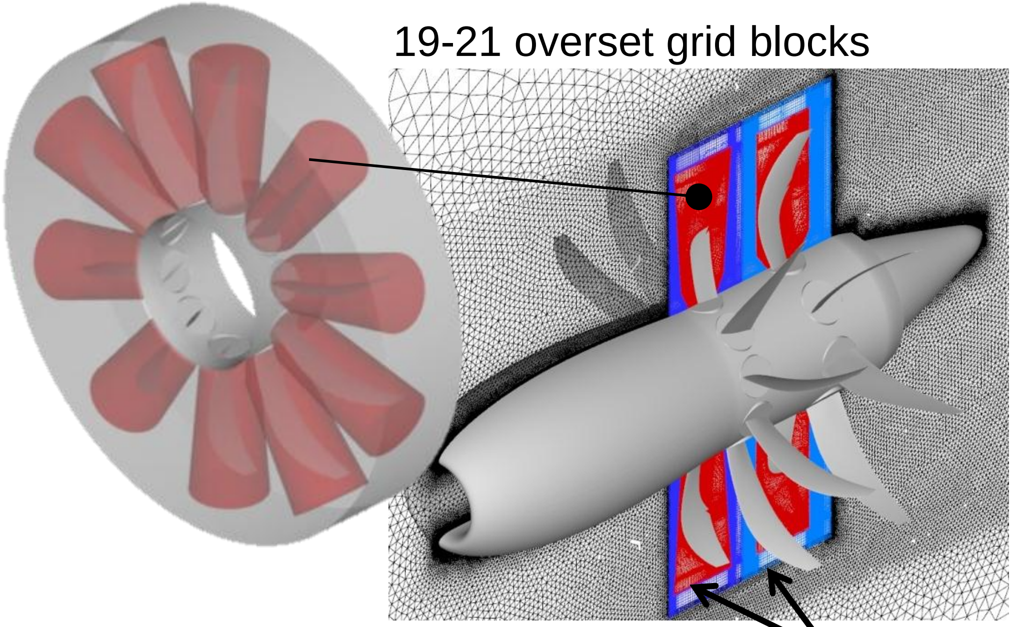
Rotations

Open Rotor Applications:
CROR Flow Analysis

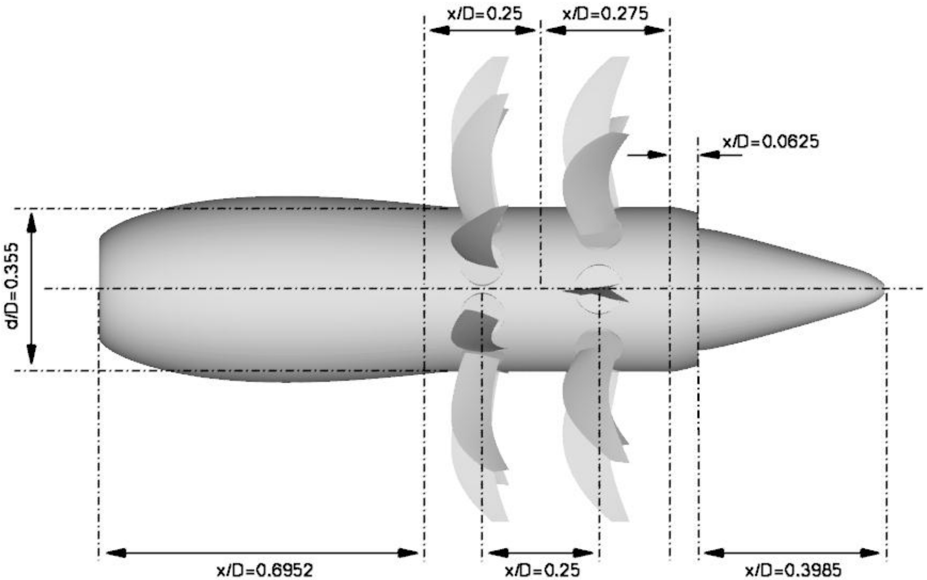
Here: Isolated rotor case

- Unstruct. background grid, struct. spinner & blade grids, „pre-generated“ holes used

19-21 overset grid blocks



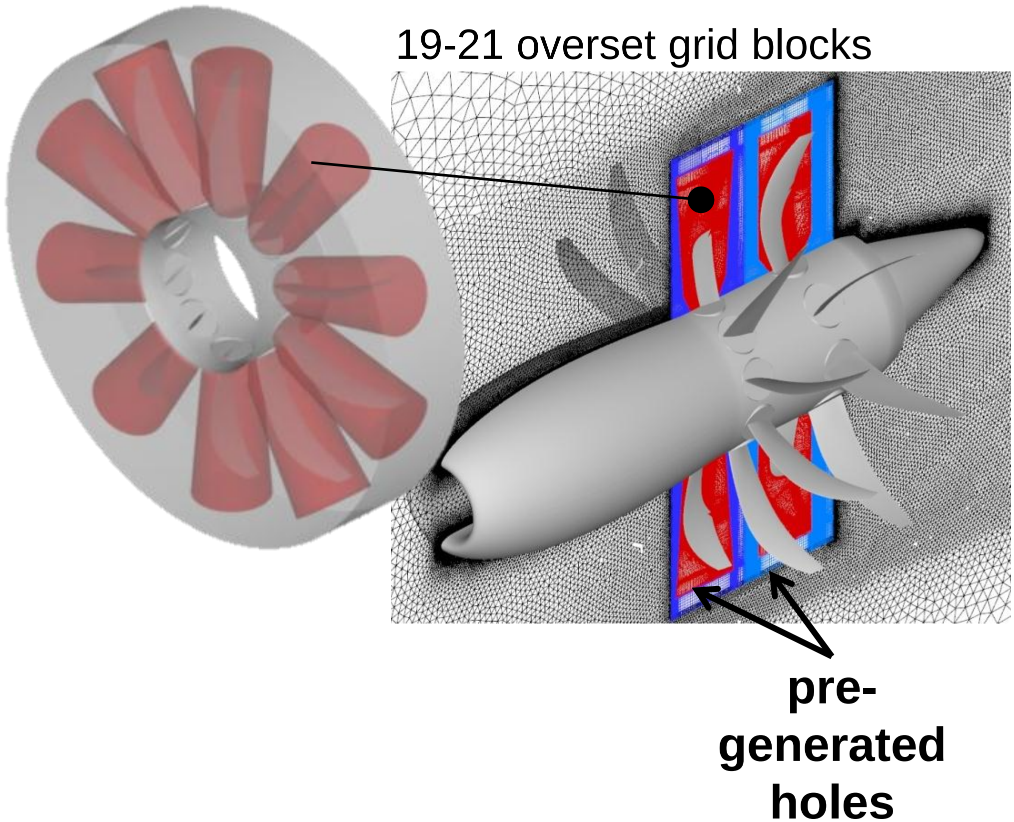
pre-generated holes

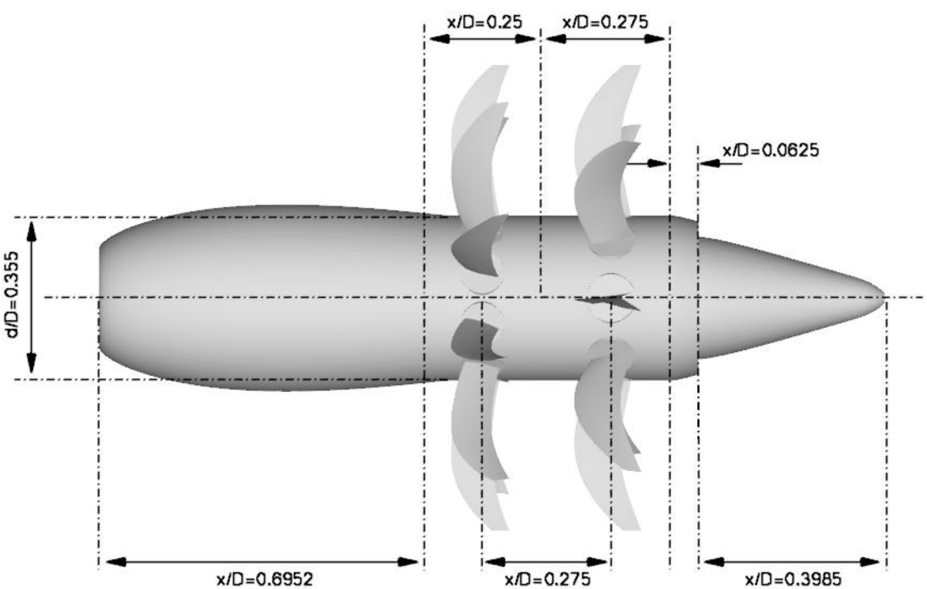


Open Rotor Applications:
CROR Flow Analysis

Here: Isolated rotor case

- Unstruct. background grid, struct. spinner & blade grids, „pre-generated“ holes used
- Perform. & noise analysis of many config. variations

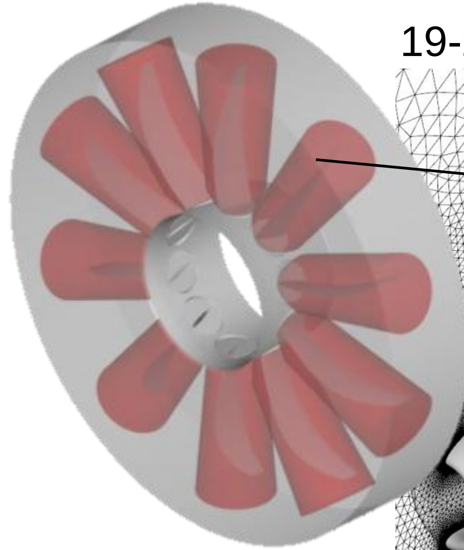




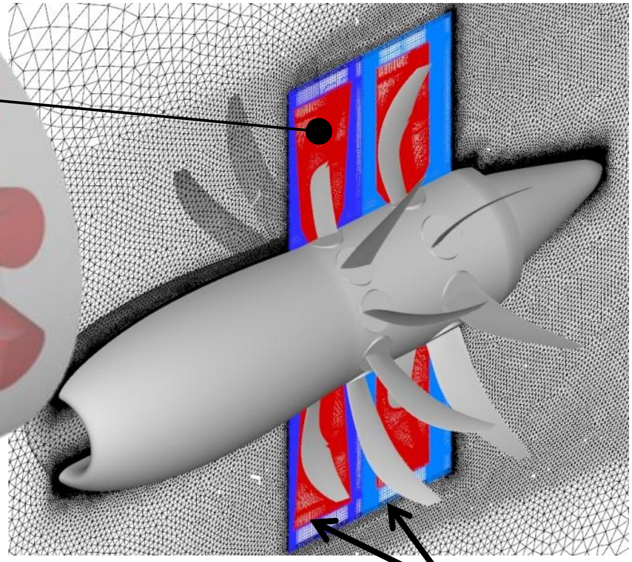
Open Rotor Applications:
CROR Flow Analysis

Here: Isolated rotor case

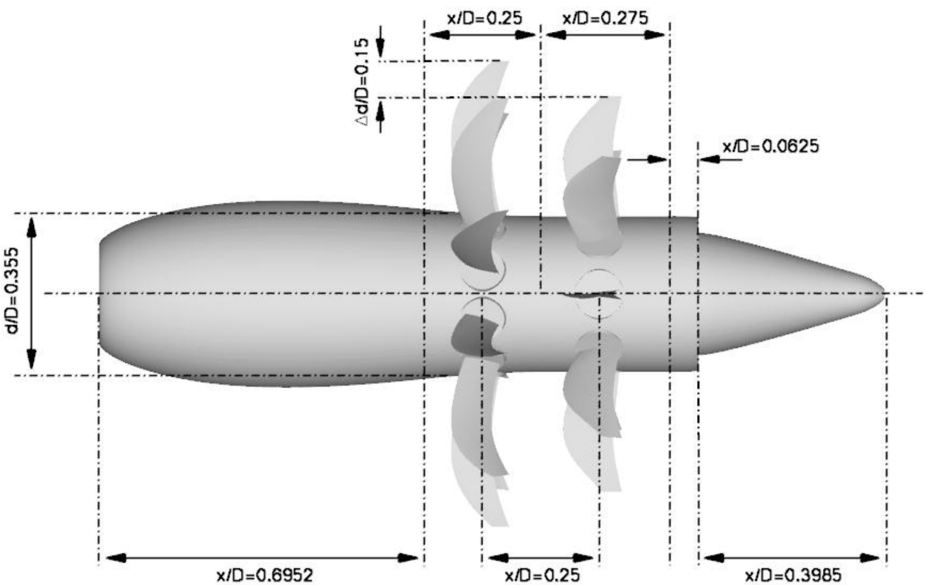
- Unstruct. background grid, struct. spinner & blade grids, „pre-generated“ holes used
- Perform. & noise analysis of many config. variations



19-21 overset grid blocks



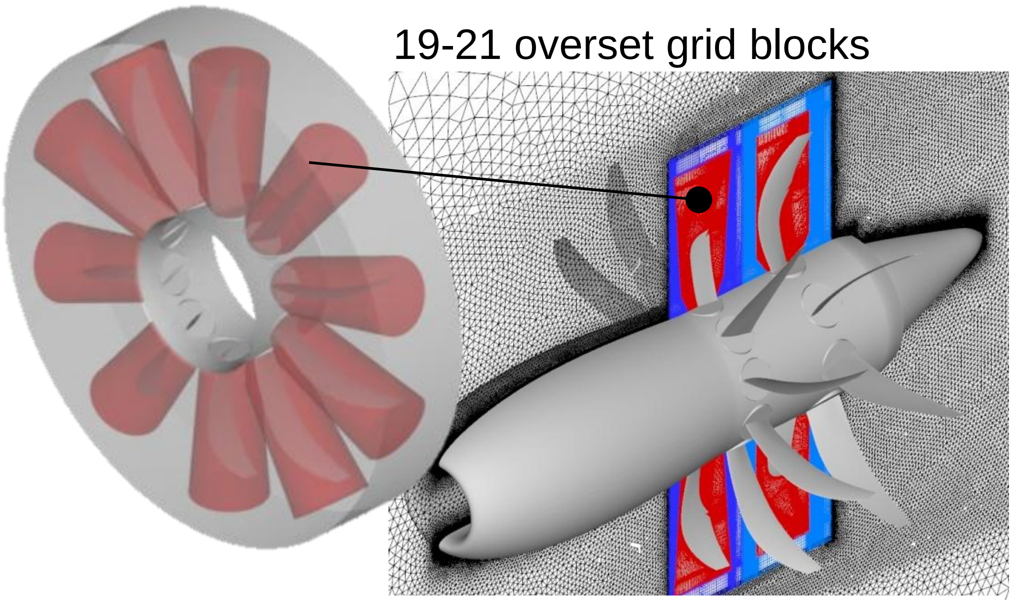
pre-generated holes

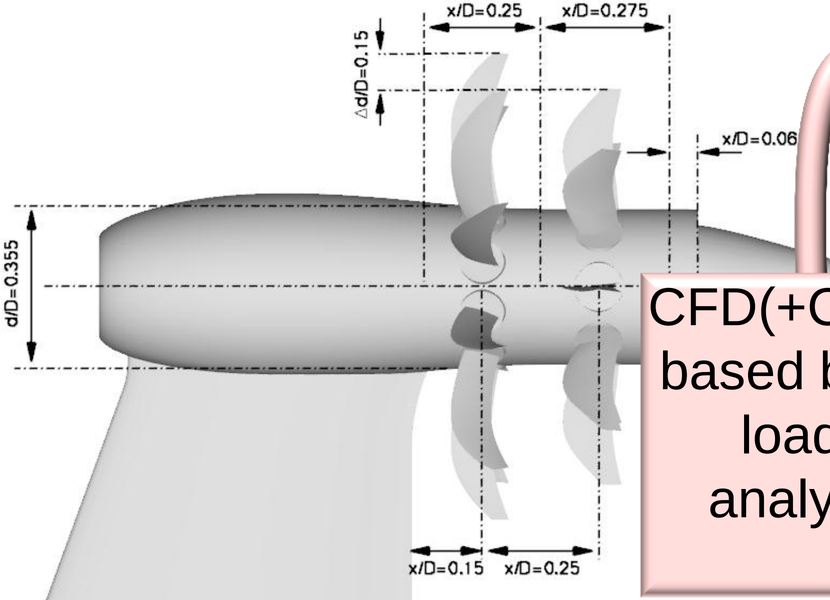


Open Rotor Applications:
CROR Flow Analysis

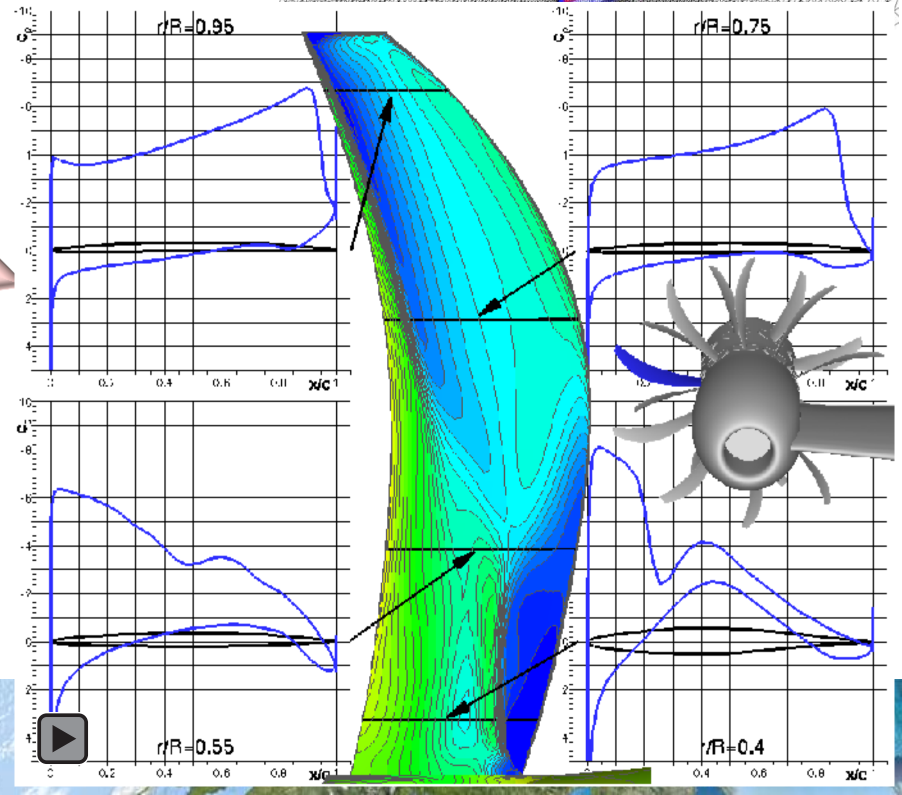
Here: Isolated rotor case

- Unstruct. background grid, struct. spinner & blade grids, „pre-generated“ holes used
- Perform. & noise analysis of many config. variations
- Comp. time: 65M nodes case, 360 procs..: 720 timesteps per revolution necessary => ~4d per rev; 6-12 revs necessary => ~1.5-3 weeks





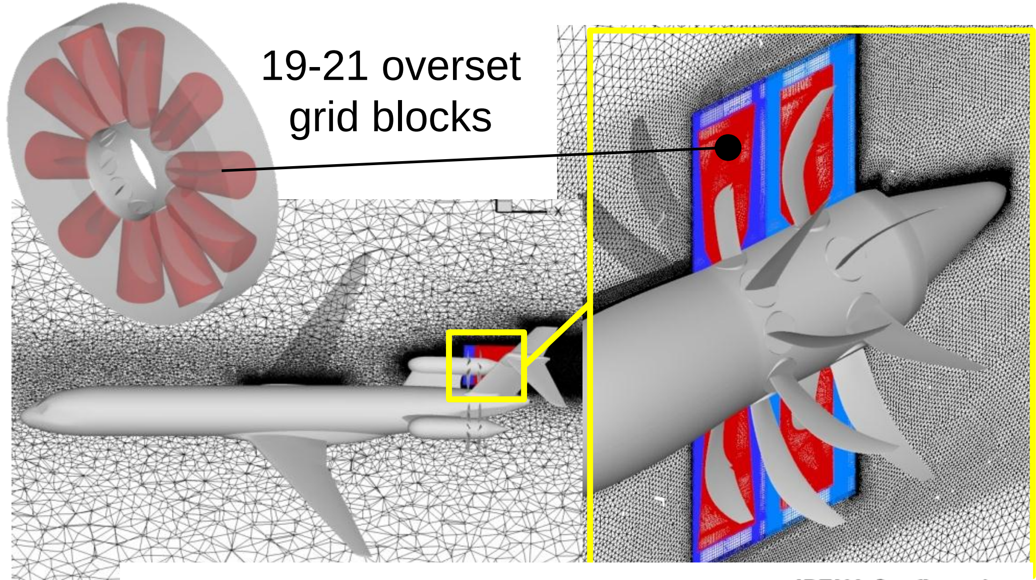
CFD(+CSM)-based blade loads analysis



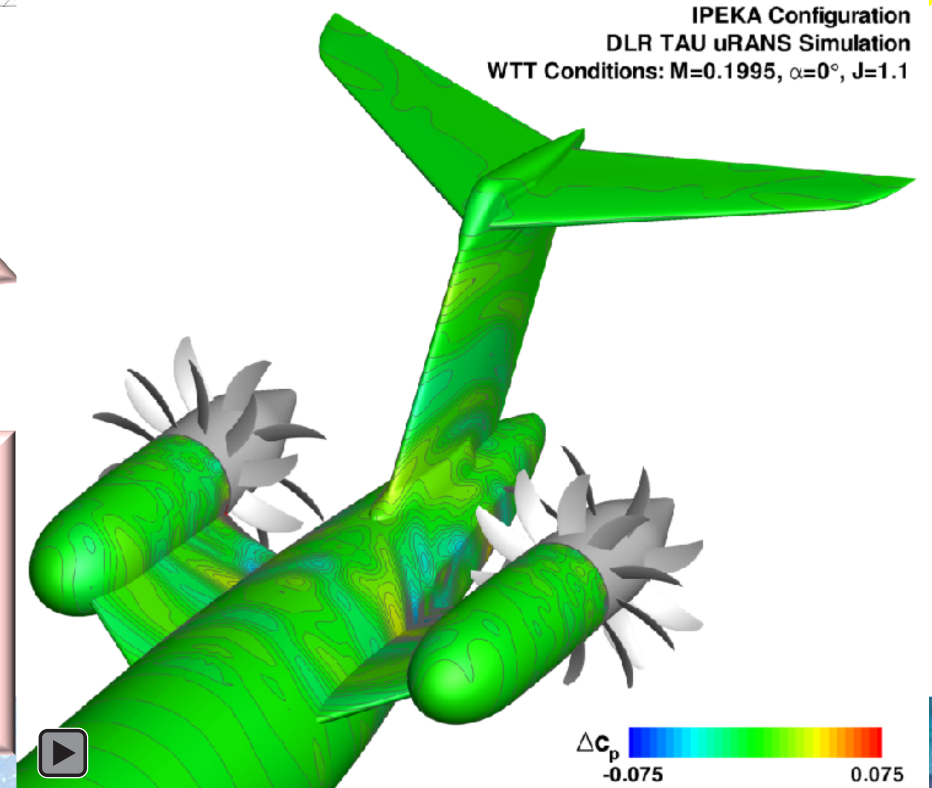
Open Rotor Applications:
CROR Flow Analysis

Here: Integrated rotor case

- Unstruct. background grid, struct. spinner & blade grids, „pre-generated“ holes used

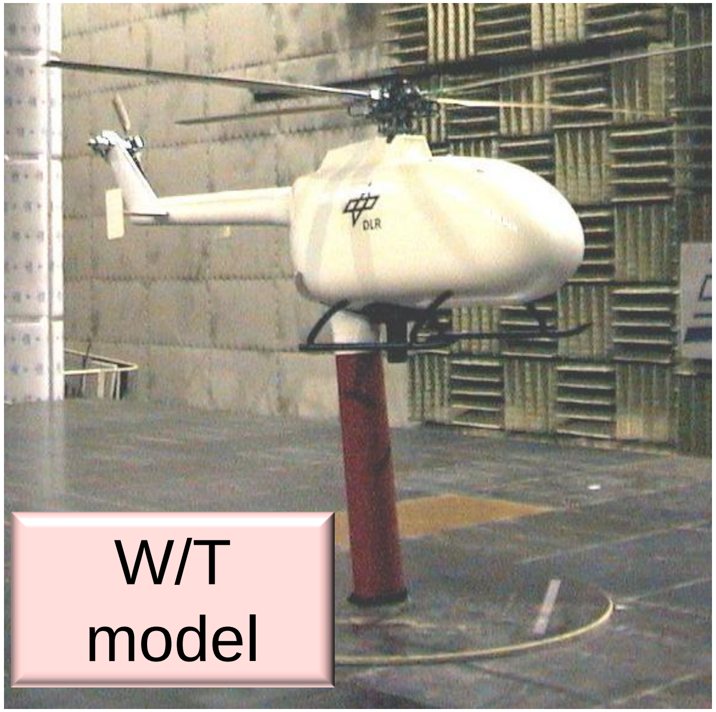


CFD-based airframe loads analysis



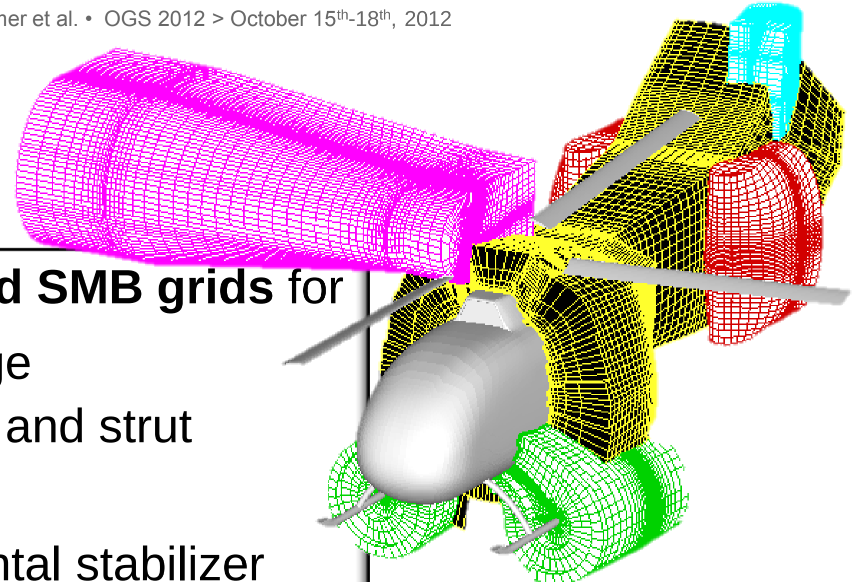
DLR

Helicopter Applications:
BO105 Case

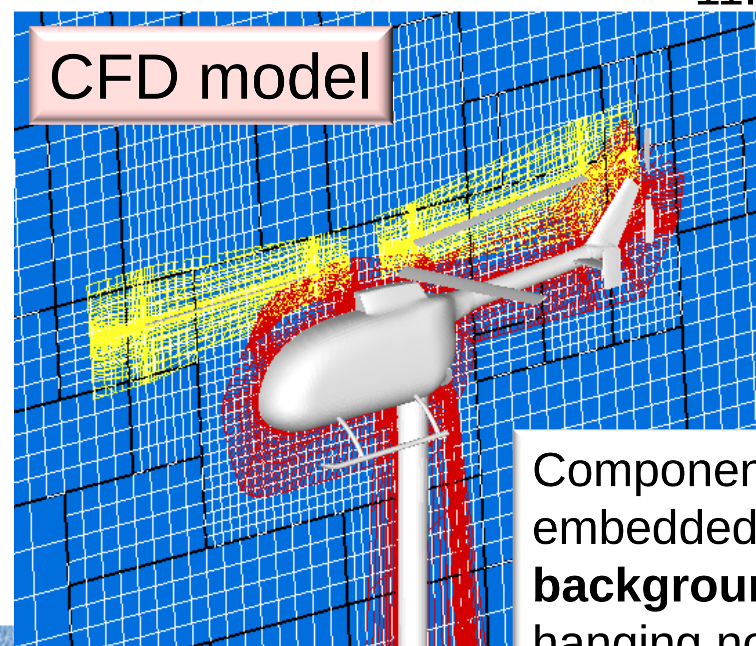


$a_{fuselage} = -5.2$
$M_{\infty} = 0.1766$
$M_{MR} = 0.652$
$M_{TR} = 0.63$
$Q_{MR} = 10.5° - 6.3° \sin(Y)$
$+ 1.9° \cos(Y)$
$Q_{TR} = 8.0°$

Nearfield SMB grids for
- fuselage
- spoiler and strut
- skids
- horizontal stabilizer
- main rotor and tail rotor



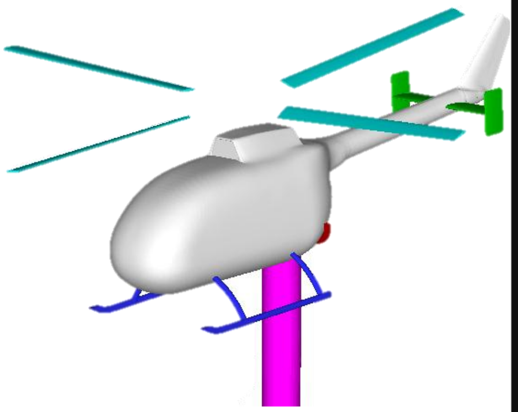
Total: 480 blocks, 11.8m nodes

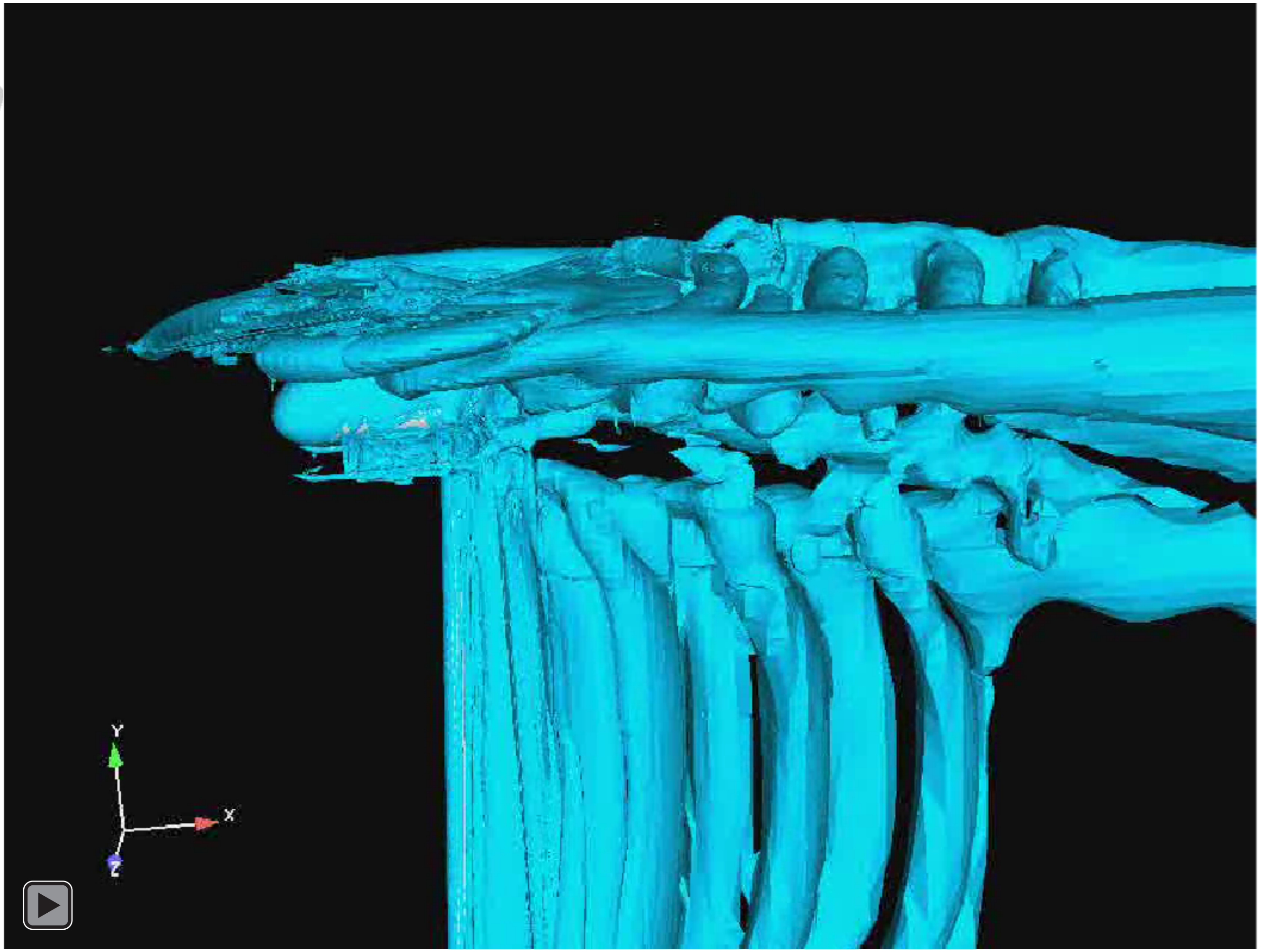


Component grids are embedded in **Cartesian background grid** with hanging nodes

Helicopter Applications:
BO105 Case

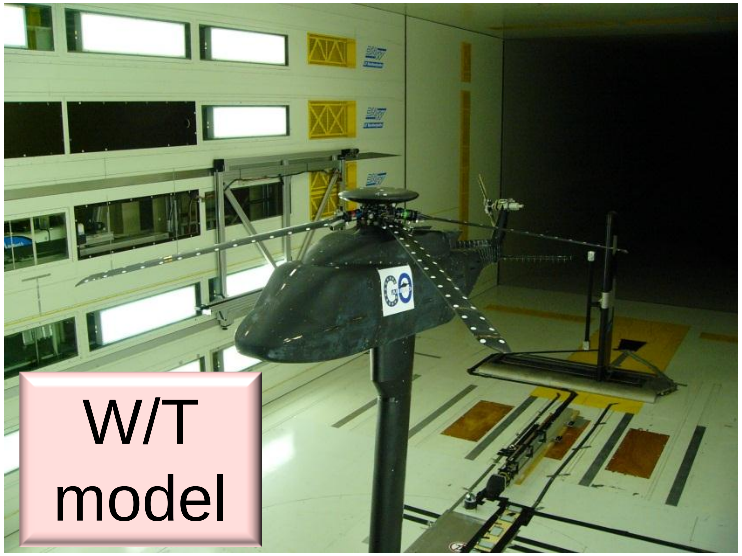
λ_2 iso-surfaces

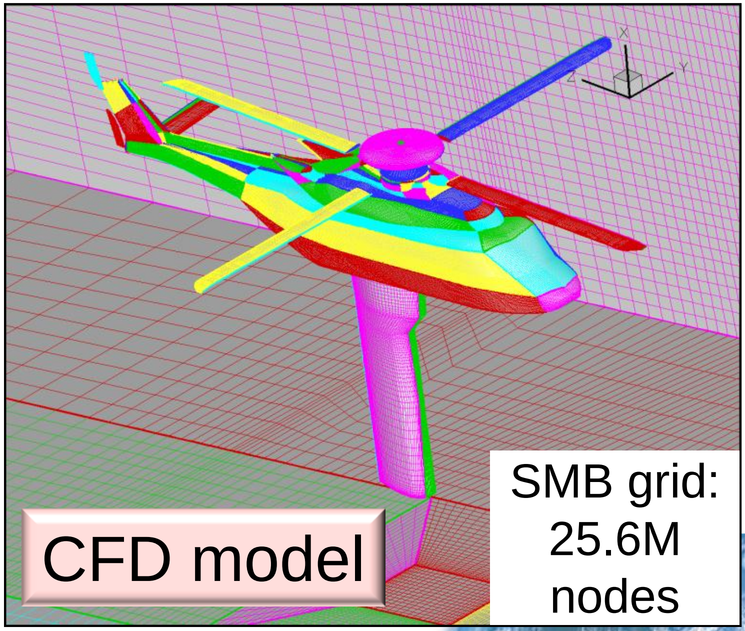


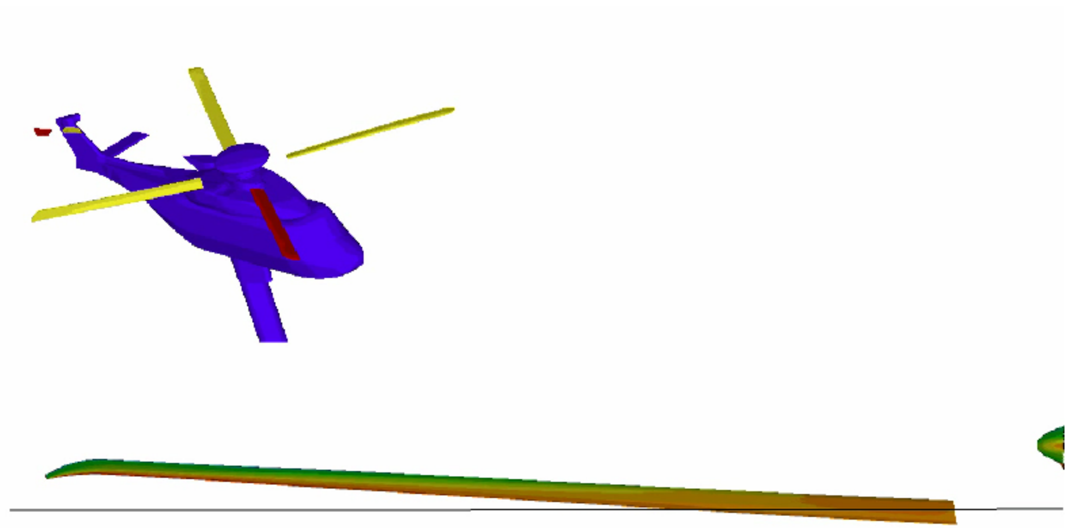


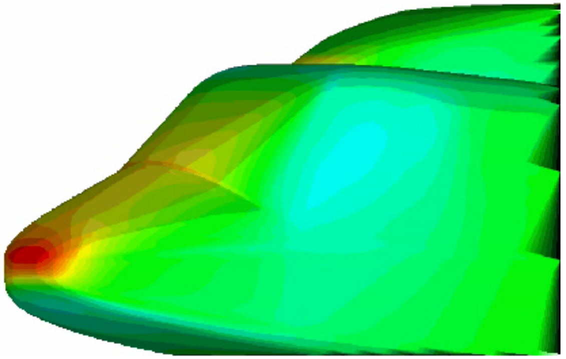
Helicopter Applications: **GOAHEAD Case**

FLOWer-HOST* loose coupling
=> trim + elastic blade defo. considered





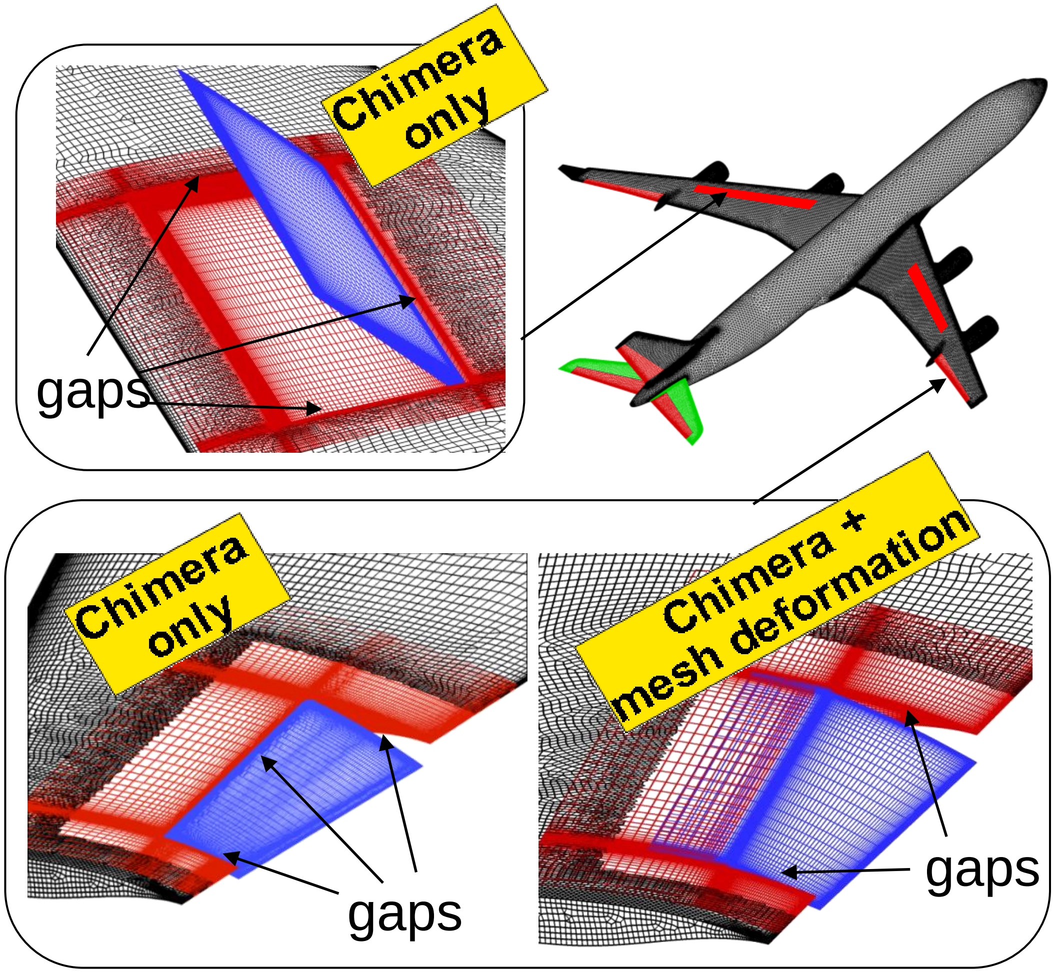




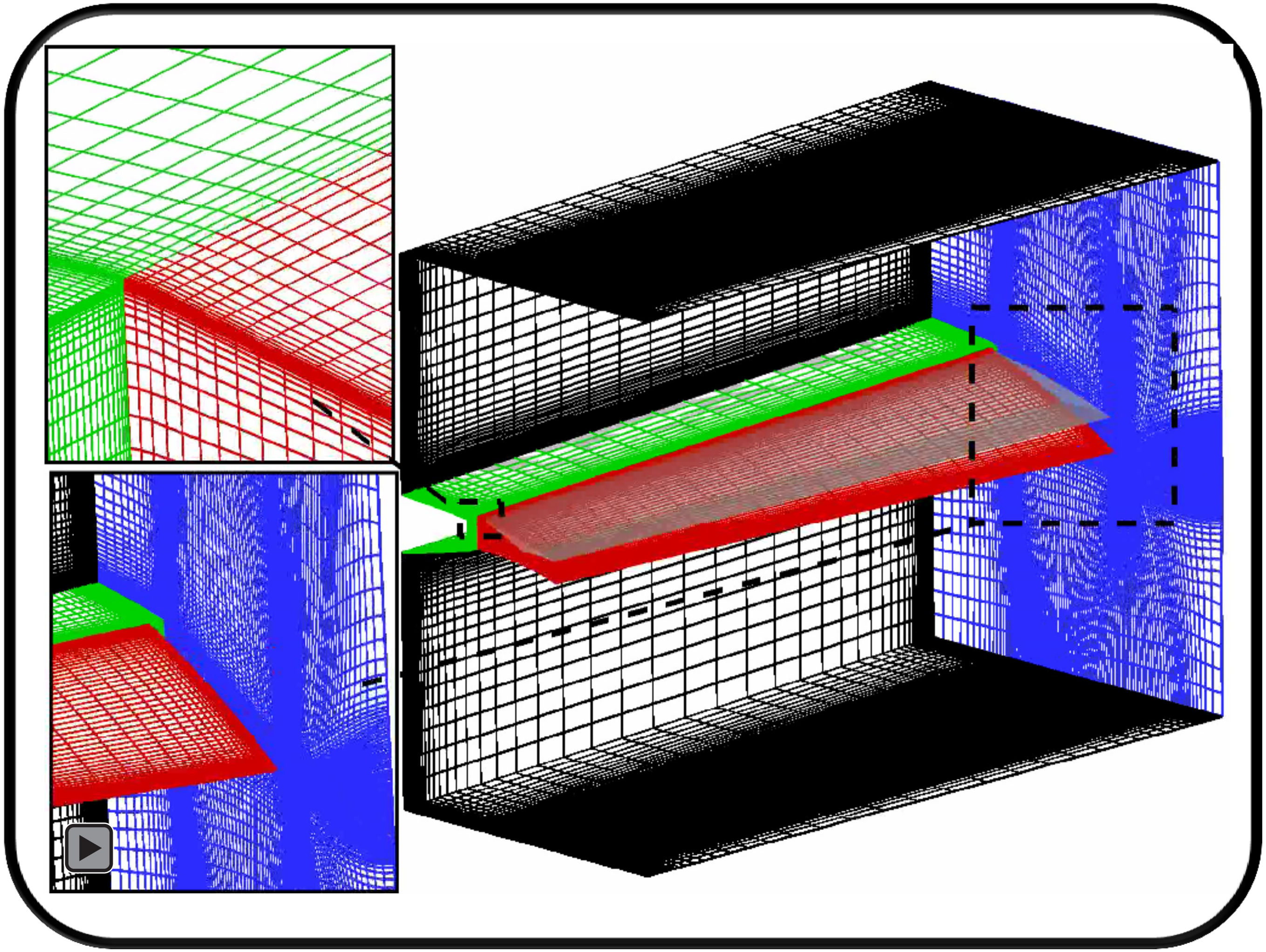
* HOST= Comprehensive analysis codes

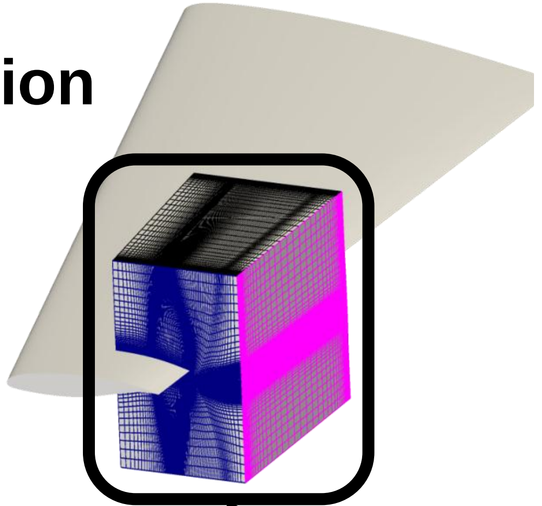
Control Surface (CS) Motions

- Meshing & sim. of configs. with **statically deflected CS** is standard application
- Though it still challenging consideration of **moving CS** becomes more important (e.g. analysis of maneuver loads & effectiveness of gust load alleviation systems)
- Main issue: Gap handling
  2 Approaches:
  - Chimera-only: Rotating Chimera block
  - Chimera+Mesh defo.: Fixed Chimera block (seems most promising)



Control Surface Motions:
Overset Grid Technology + Mesh Deformation

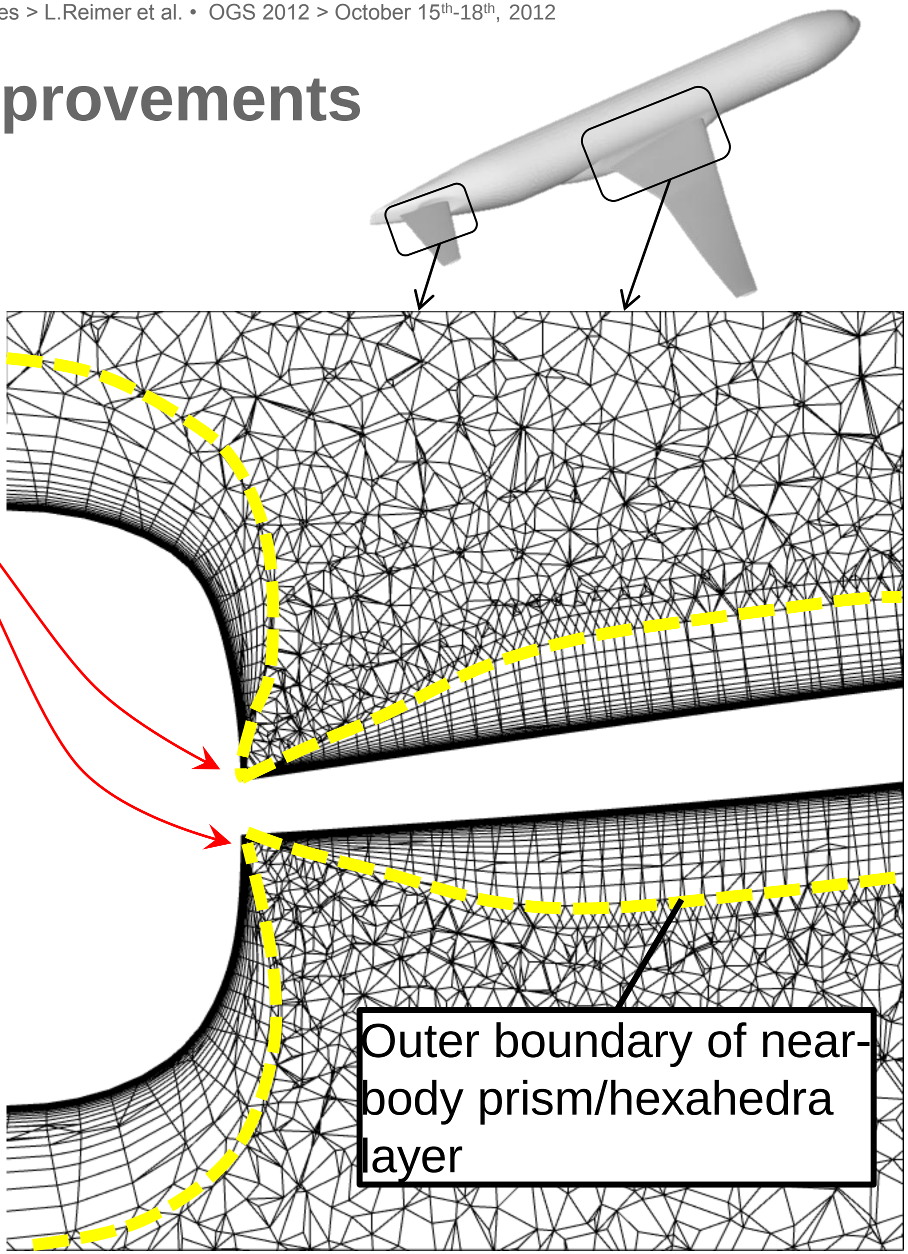




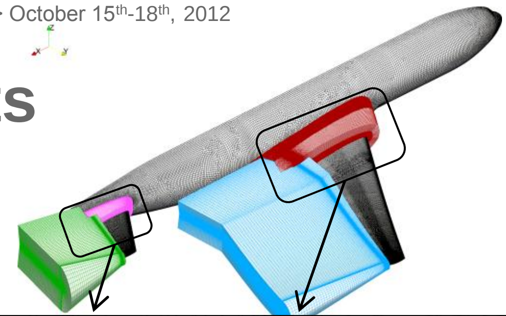
Mesh defo. only applied to Chimera block containing the aileron

Local Discretisation Improvements
Example from DPW-4

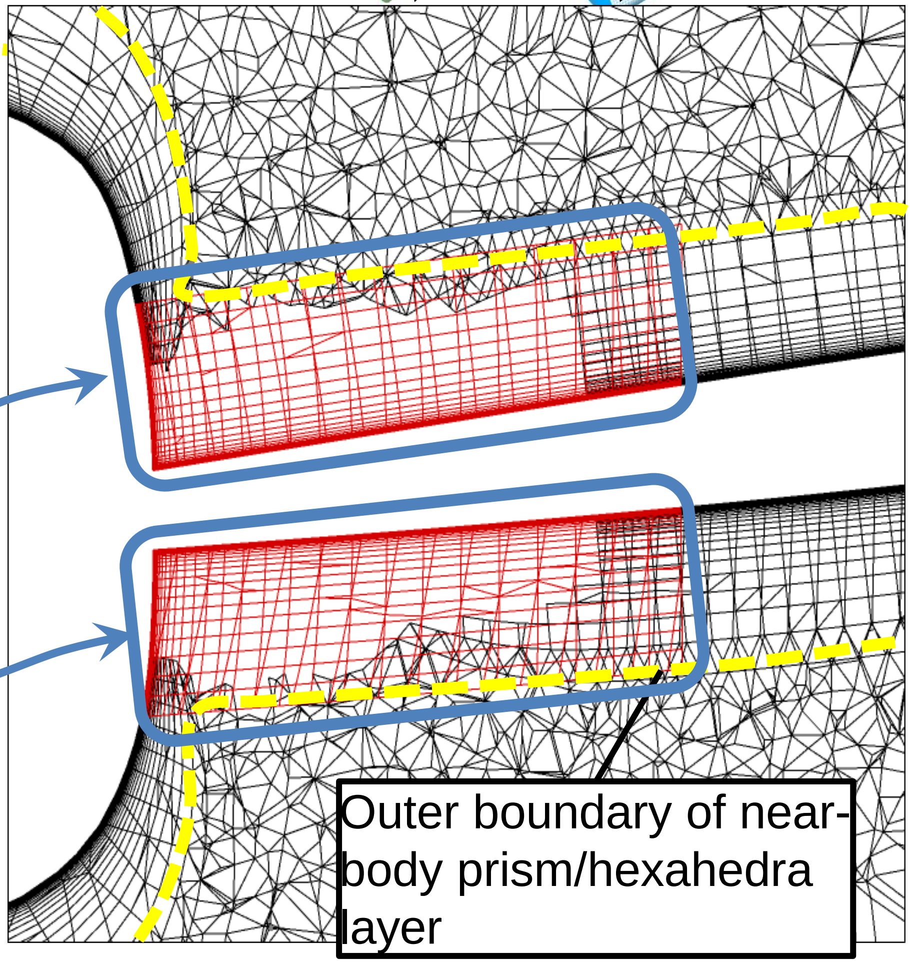
- Boundary layer in wing/HTP-fuselage junction of inappropriate quality
- Flow separations might not be detected at all or are metric driven



Local Discretisation Improvements
Example from DPW-4

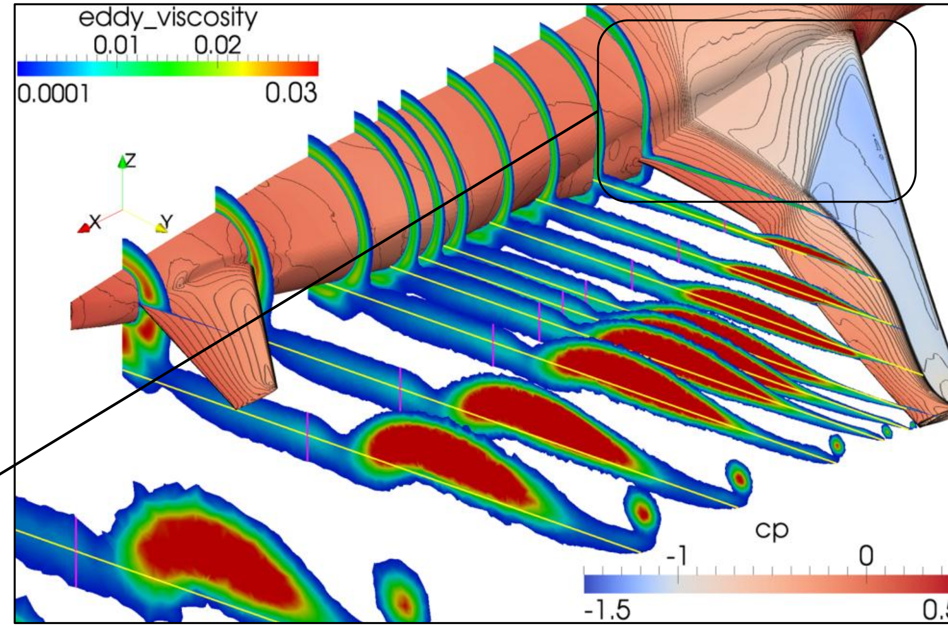


- ~~Boundary layer in wing/HTP-fuselage junction of inappropriate quality~~
- ~~Flow separations might not be detected at all or are metric driven~~
- Discretisation can be improved locally by inserting a structure Chimera grid

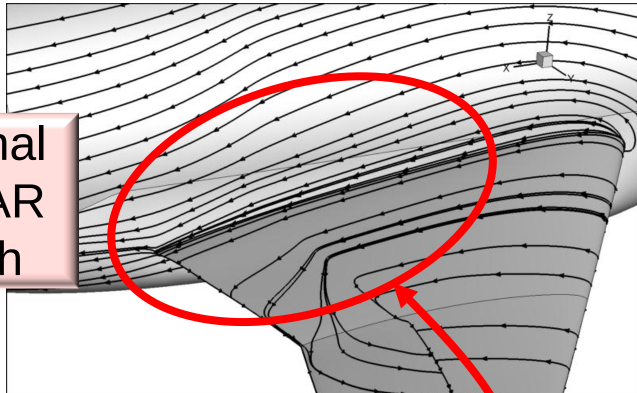


Local Discretisation Improvements

Example from DPW-4

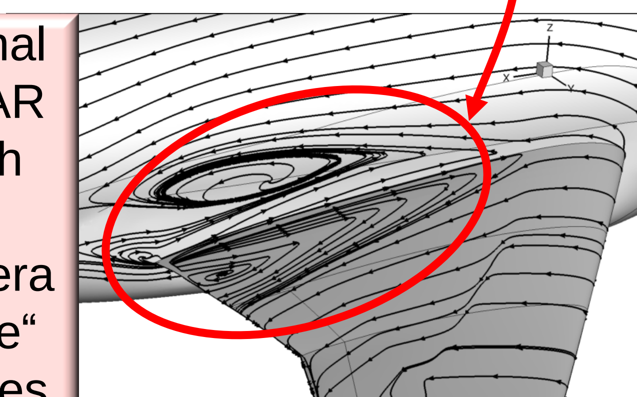


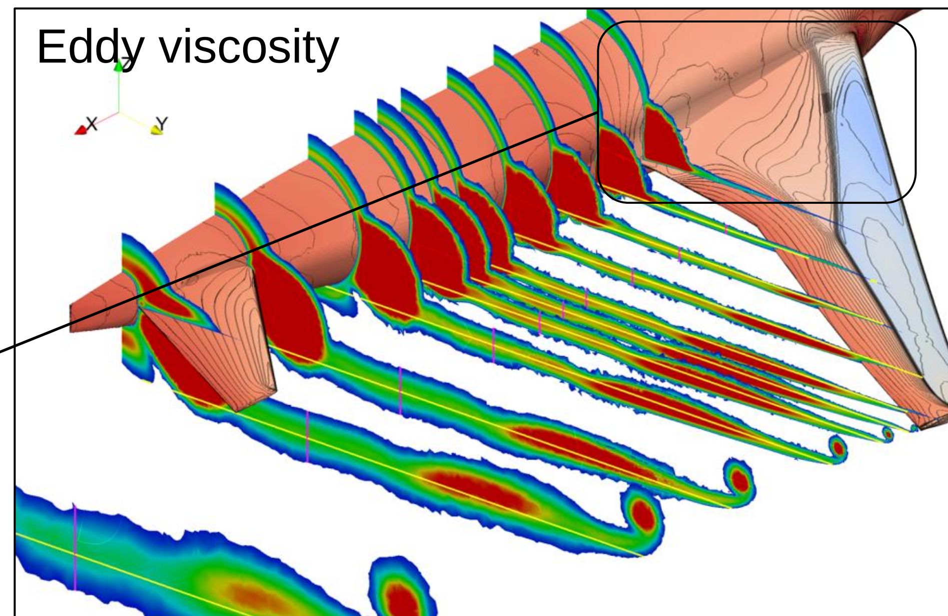
Original SOLAR mesh



streamlines

Original SOLAR mesh + Chimera „glove“ meshes

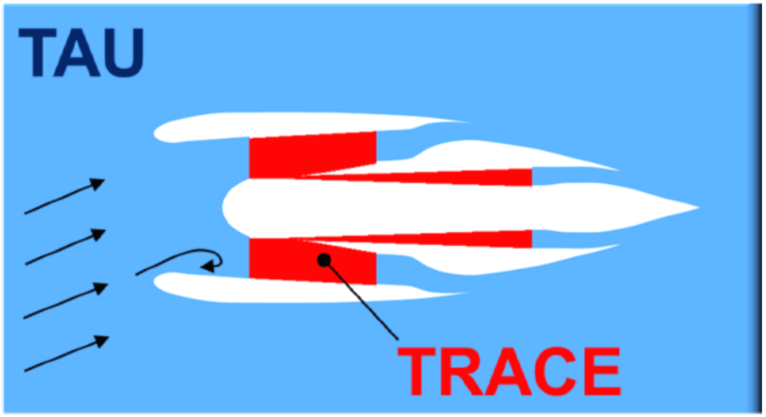


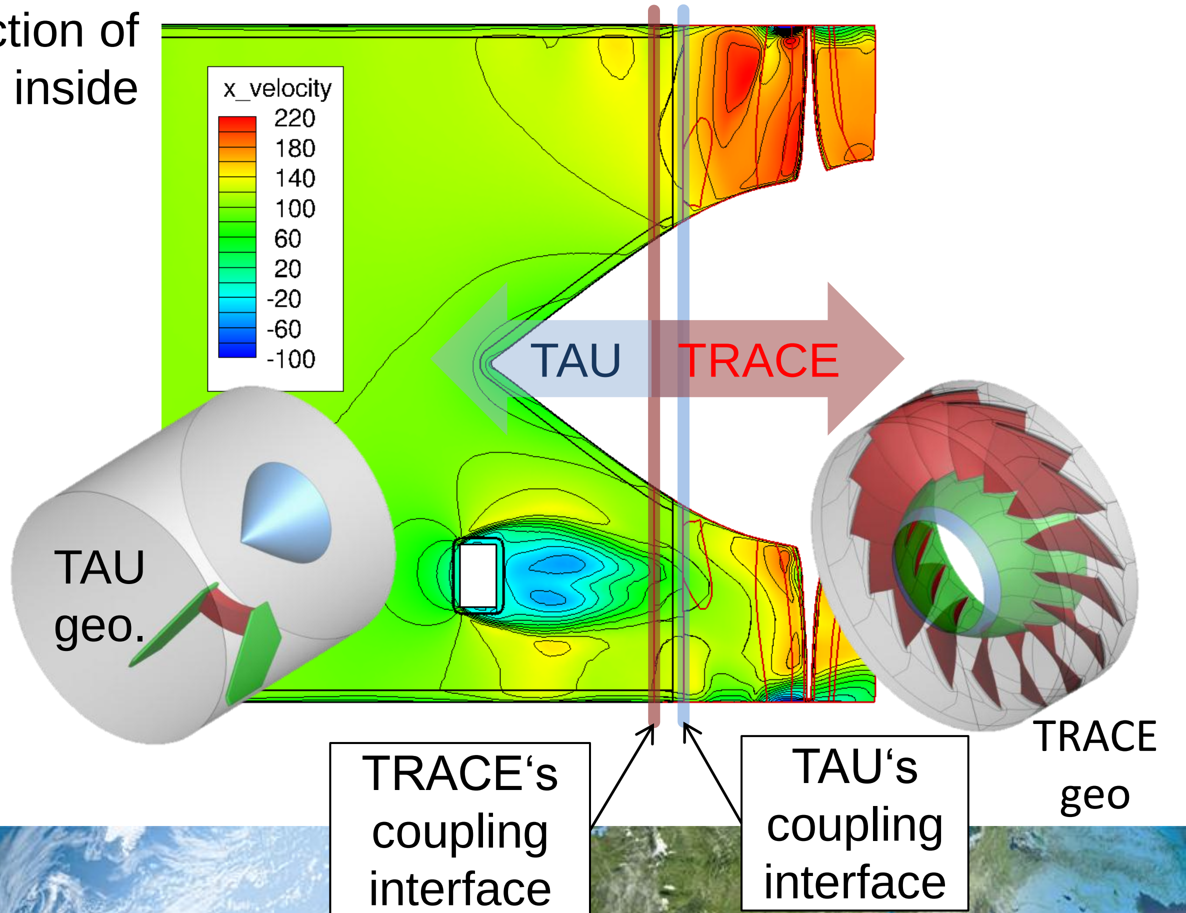


DLR

Code-to-Code Coupling: **TAU-TRACE**

- **Objective**: Flow simulation of a stalling nacelle including entry of the separated flow into the engine fan
- Requires accurate flow prediction of both external flow and flow inside engine fan
- **TRACE**:
  - DLR‘s turbomachinery CFD code
  - Multi-block-structured
  - cell-centered code (TAU: node-centered)
- Coupling controlled by socket communication layer
- Spatial coupling with Chimera





Code-to-Code Coupling:
TAU-Cartesian TAU

Problem: 2nd order spatial accuracy too dissipative for accurate computation of convection of gusts or wake vortices

Solution: Coupling of higher-order Cartesian solver (CTAU) to 2nd order solver (TAU)

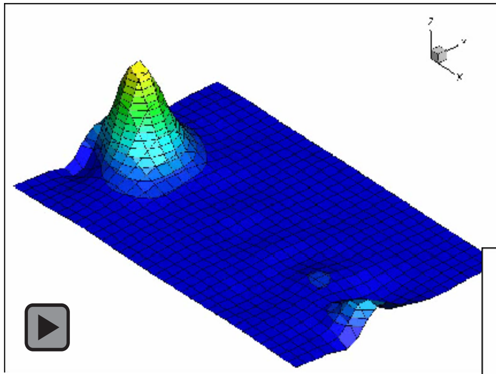
Cartesian TAU:
- ≥ 4th order spatially accurate (Pade scheme)
- very coarse Cartesian meshes, easy to generate

Conventional TAU:
- 2nd order spatially accurate
- fine body-fitted hybrid meshes

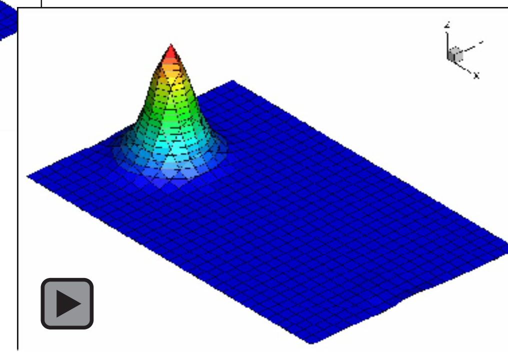
Coupling: overset grid interpolation

Example: Ehrenfried vortex

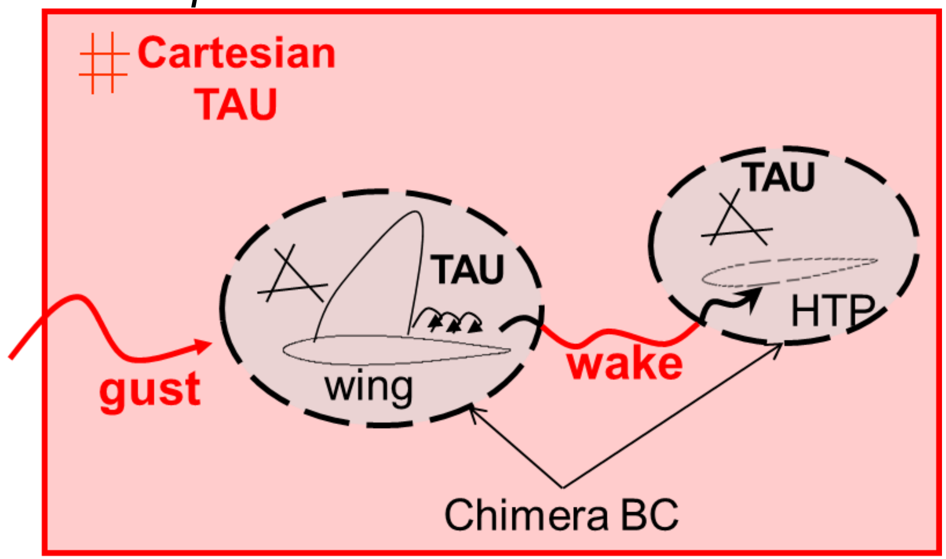
2nd order central



4th order Pade



Concept:



Conclusions

- DLR's CFD solvers are very mature, feature standard Chimera capabilities and are thus applicable to wide range of industrially-relevant configurations that need Chimera
- Our current major problem:
  - Too many manual operations needed to setup Chimera cases
  - Lack of toolkits like CGT, etc., at DLR
  - High demand for a more „automated" Chimera case setup process
- Since there is now a fast growing spectrum of applications that rely on the Chimera technique (Digital-X project: -> virtual flight testing), we currently increase our effort to bring forward our Chimera capabilities (e.g. overlap optimization, overlap adaptation, further development of CTAU solver, …)